ORNL-5967
Dist. Category UC-11, 41

An Analysis/Plot Generation Code with Significance Levels Computed Using Kolmogorov-Smirnov Statistics Valid for Both Large and Small Samples

S. E. Kurtz
Spring Arbor College
Spring Arbor, Michigan 49283

D. E. Fields
Health and Safety Research Division

Date Published - October 1983

Prepared by the
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
U.S. DEPARTMENT OF ENERGY
Office of Defense Waste and Byproducts Management
Under Contract No. W-7405-eng-26

DISCLAIMER

CONTENTS

ACKNOWLEDGEMENTS

The authors wish to thank C. W. Miller and M. Cristy for their review of this report and for their helpful comments. Other valuable suggestions were received from C. F. Baes III and F. O. Hoffman. We are also grateful to S. Norman for assisting in preparing the manuscript for publication. Publication of this work was supported by the Office of Defense Waste and Byproducts Management, United States Department of Energy, under the Development of Low Level Waste Disposal Facilities task, FTPA # ONL-WN13 and Activity # AR 05 10 05 K.

ABSTRACT

This report describes a version of the TERPED/P computer code that is very useful for small data sets. A new algorithm for determining the Kolmogorov-Smirnov (KS) statistics is used to extend program applicability.

The TERPED/P code facilitates the analysis of experimental data and assists the user in determining its probability distribution function. Graphical and numerical tests are performed interactively in accordance with the user's assumption of normally or log-normally distributed data. Statistical analysis options include computation of the chi-square statistic and the KS one-sample test statistic and the corresponding significance levels. Cumulative probability plots of the user's data are generated either via a local graphics terminal, a local line printer or character-oriented terminal, or a remote high-resolution graphics device such as the FR80 film plotter or the Calcomp paper plotter.

Several useful computer methodologies suffer from limitations of their implementations of the KS nonparametric test. This test is one of the more powerful analysis tools for examining the validity of an assumption about the probability distribution of a set of data. KS algorithms are found in other analysis codes, including the Statistical Analysis Subroutine (SAS) package and earlier versions of TERPED. The inability of these algorithms to generate significance levels for sample sizes less than 50 has limited their usefulness. The release of the TERPED code described herein contains algorithms to allow computation of the KS statistic and significance level for data sets of, if the user wishes, as few as three points. Values computed for the KS statistic are within 3% of the correct value for all data set sizes.

1. INTRODUCTION

This report describes a modification of the TERPED/P code having significantly improved analytical capability. Applicability of the code to small data sets is extended considerably over earlier versions of the code and over SAS[1] or IMSL[2] codes.

TERPED/P is a version of the TERPED interactive statistical analysis and plotting program with modifications to provide for printer-plotter output of graphical data. TERPED/P (Glandon and Fields, 1981) and its predecessors, TERPED (Fields, 1981) and TEDPED (Fields, Little and Shaeffer, 1980), were designed to assist in determining whether a set of data may be considered to be normally or log-normally distributed by performing various numerical and graphical tests. Knowledge of the nature of the probability distribution is useful for the evaluation and assessment of environmental standards and impacts, interpretation of the output of mathematical models (Fields and Glandon, 1981) and interpretation of environmental radiation data.

All three programs were designed to operate in an interactive mode. TEDPED required a graphics terminal for output of the graphical analysis. TERPED generated plot data sets through the use of DISSPLA[3] subroutines which could be viewed on the graphics terminal or routed to a high-resolution plotter, such as a Calcomp or FR80 film plotter. TERPED/P, with its printer-plotter subroutines, provided a form of graphical output which is independent of the DISSPLA routines not accessible to many users.

The TERPED/P code provided for the evaluation of Kolmogorov-Smirnov (KS) one sample goodness-of-fit test. IMSL subroutines were used to perform the test, but the IMSL routines give valid results only when $N \geq 80$, where N is the sample size. However, many data sets of interest

1. Software distributed by SAS Institute, Inc.

2. Software distributed by ISSCO (Integrated Software Systems Corporation).

3. Software distributed by International Mathematical and Statistical Libraries, Inc.

have fewer than 80 data points. The purpose of the modifications described herein is to overcome the $N \geq 80$ constraint.

The present version of TERPED/P provides the user with a numerical and graphical description of a set of data related to an assumption of either normal or log-normal distribution. The plots produced, either high resolution or printer-plotter, are of linearized cumulative probability with a fitted line showing how well the set of data fits the assumed distribution. Various statistical parameters are computed and optional tests generate the KS and the chi-squared statistics. Data can be input by hand or from a disk file. The code is written in FORTRAN-10 for execution on a DEC SYSTEM-10 computer. Typical CPU execution time for a data set with a value of 50 or fewer is 0.32 s, exclusive of plotting time. The size of the code is approximately 1600 card images.

TERPED/P may be useful even to SAS users. While both provide printer-plots, SAS cannot produce high resolution plots. For the Kolmogorov-Smirnov tests (KS test), SAS requires $N \geq 50$ for valid results. Also, SAS is designed for batch processing while TERPED/P is designed for a file oriented, time-sharing environment.

The TERPED/P code is a useful tool for determining data statistics. However, code users must recognize their responsibility to collect data with insight and interpret results with judgment.

2. PROGRAM STRUCTURE

The original version of TERPED/P used the IMSL subroutines NKS1 and MDSMR to perform the KS test. NKS1 calculated the value of the statistic, while MDSMR computed significance levels based on that statistic.

Our modifications to TERPED/P involved the replacement of these two IMSL routines with three subroutines of our own writing and the expansion of an already existing block data subprogram. The new subroutines are called NKS, NGT80 and NLE80. NKS corresponds to the IMSL routine NKS1 and calculates the KS statistic. NGT80 corresponds to MDSMR and computes significance levels when N>80, using asymptotic relations due to Smirnov (Smirnov, 1948). NLE80 has no corresponding IMSL subroutine and its purpose is to compute significance levels when $N \leq 80$ by means of a previously developed algorithm (Kurtz and Fields, 1983). The algorithm involves the interpolations of adjacent entries in a table of probabilities computed from Kolmogorov's formulas by Birnbaum (Birnbaum, 1952), which has been placed in the block data subprogram.

In addition to those subprogram modifications, a number of structural modifications were made throughout the code so that the new subroutines would fit in properly and smoothly.

The flow of program control has changed very little from that of the original TERPED/P code. The only change is that rather than NKS1 calling MDSMR only, NKS will call either NGT80 or NLE80. NKS is called in the same manner as NKS1.

This version the TERPED/P program consists of 25 subprograms and a number of other subroutines that comprise part of the DISSPLA graphics package. The program structure is shown in Fig. 1. Provision has been made for two methods of data entry. Statistics are output to the computer terminal as they are computed. Graphical outputs are user selectable and consist of a printer-plotter and a high-resolution plotter.

A FORTRAN listing of TERPED/P is included in Appendix C. The Main Program structures flow of control. It also accepts some control information and performs intermediate calculations for computing the correlation between input data and the corresponding hypothetical distribution.

ORNL - DWG 81-10028

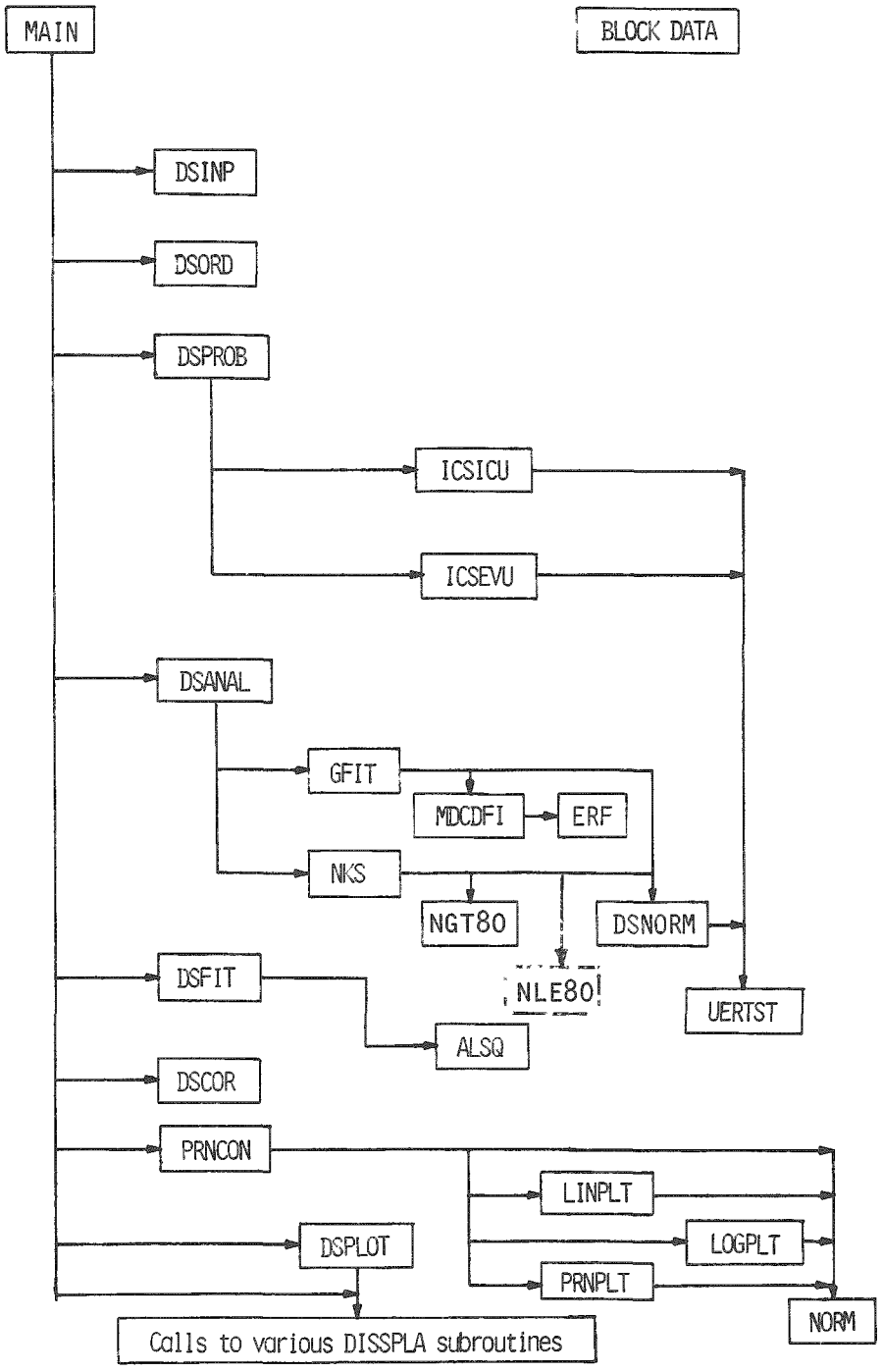

Figure 1. Program structure of the revised TERPED/P code.

Subroutine DSINP prompts the user to input program control information and data. Data can be entered directly into the program by using the terminal keyboard or can be read from a file that has been stored on disk. The file must be named "FORxx.DAT" where xx is between 07 and 63, inclusively. The sample run in Appendix A documents use of a data file.

The data are ranked according to magnitude by subroutine DSORD. Subroutine DSPROB computes a cumulative probability for each data value. It also performs a cubic spline fit to a table of error function arguments stored in the BLOCK DATA subprogram to compute the corresponding linearized cumulative probability. Subroutine ICSICU* and ICSEVU* are called by DSPROB to determine the spline coefficients near the data point and to evaluate the linearized probability, respectively.

Subroutine DSANAL performs preliminary data analysis and if the user has called for a chi-squared test or KS test, calls GFIT* or NKS, respectively. GFIT uses subroutine MDCDFI* to compute the probability for rejection of the null hypothesis and in turn calls ERF* to compute the value of the error function. NKS invokes either NLE80 or NGT80, the selection being based on the sample size. NLE80 computes the significance level if there are 80 or fewer data points and NGT80 computes the significance level if there are more than 80 points. Both GFIT and NKS use subroutine DSNORM which is the theoretical probability distribution function against which the data is tested. All of the IMSL* subroutines call subroutine UERTST* which generates and prints warning and error messages which are listed in Table 1.

Subroutine DSFIT performs a linear least-squares fit to the data points. Subroutine ALSQ (Westley and Watts, 1970), which determines the coefficients of the fit, is called by DSFIT.

Pearson's correlation coefficient between data points and corresponding points along the linear least-squares fit gives a measure of the validity of normal or lognormal hypothesis. The correlation coefficient is calculated by subroutine DSCOR, but is printed by the main program.

---

* Software written by International Math and Statistical Libraries, Inc., Houston, Texas.

Table 1. List of error messages generated by TERPED/P

Message	Comments
A) Number of points cannot exceed 300	TERPED arrays are dimensioned to accept up to 300 data points. The user must redimension for data sets greater than 300 points.
B) Invalid input; try again	The user gave an inappropriate response to the prompt message.
C) Error from ICSICU, from DSPROB: Data Point J. ERR=I	Evaluation of whether the cubic spline coefficients about the Jth data point has been successful. For NX=number of points, less 1, and for IC=row dimension spline coefficient matrix; ERR=129 indicates IC < (NX-1). ERR=130 indicates NX < 2. ERR=131 Indicates data points incorrectly ordered. For an evaluation of the severity of these errors, see E).
D) Error from ISCEVU, DSPROP; Data Point J, ERR=1	Evaluation of the linearized probability for the Jth data point has been unsuccessful. ERR=33 indicates the calculated abscissa of the cubic spline is less than the minimum abscissa represented in the data. ERR=34 indicates the calculated abscissa of the cubic spline is greater than the maximum abscissa represented in the data. For comments on the severity of these errors, see E).
E) ***IMSL(UERTST)*** (Warning type) Name IER=K	(Warning type) is either "warning," "warning with fix," "terminal" error, or "nondefined." NAME identifies the subroutine in which the error was generated. This message supplements C) and D). The value of K is computed, given I defined as in C) and D), as follows: K=I+32 (warning) =I+64 (warning with fix) =I+128 (terminal error).

Subroutine PRNCON initializes a matrix containing printer-plotter information and calls subroutine LINPLOT to plot data according to a normal hypothesis or subroutine LOGPLT, which plots data when the log-normal hypothesis has been selected. PRNCON, LINPLT, and LOGPLT use subroutine NORM to normalize data according to use input scaling values. Subroutine PRNPLT outputs the plot to the user-specified device. The 'F' character represents the fitted point, the 'D' character represents a data point, and a '*' represents fit and data point coincidence.

Subroutine DSPLOT plots in the input data and the results of the linear least-squares fit on a user-defined ordinate with a cumulative probability-derived abscissa. The plot is high quality/resolution and can be routed to a Calcomp paper plotter or FR80 film plotter. DSPLOT uses several DISSPLA subroutines.

3. DISCUSSION OF CODE USAGE

The enhancements to TERPED/P result in no significant changes in code usage from the original version. All data input is the same and the output differs only when the sample contains fewer than 80 data points. In the case of the smaller sample sets no warning message is printed as in the original version.

TERPED/P is interactive and produces many prompting messages which the user should find self-explanatory. Although the sample size warning is no longer applicable, the code generates a number of other warning and error messages, some of which are abbreviated. These messages are summarized in Table 1.

Appendix A contains the computer-user dialog for a complete run of the TERPED/P program. In the sample dialog shown in this Appendix, the user first assigns his terminal the device number 6 by executing the command "ASS TTY 6." He then begins execution of the program and provides access to various plotting utilities with the command "EX TERPED.P, @ SYS:DISPOP." The SYS:DISPOP portion of the command permits linkage to DISSPLA subroutines. The user then employs the code to analyze a set of data. First, a normal hypothesis is selected with a Kolmogorov-Smirnov goodness-of-fit test. Next, a printer plot is generated to visually check the data and a high-resolution compressed plot data set is generated. The user may then restart the code and select the lognormal hypothesis. A printer plot is then generated for a visual check and finally a compressed plot data set may be generated. The user then exits the code and checks the file directory for the plot data file (extension .POP). A text summary of the plot is generated concurrently with the plot data set and is automatically sent to a line printer. The disk file may be deleted after exiting of the program.

REFERENCES

Z. W. Birnbaum, "Numerical Tabulation of the Distribution of Kolmogorov's Statistic for Finite Sample Size," American Statistical Assn. Journal 47, 425 (1952).

D. E. Fields, TERPED: A Versatile Code for Examining the Distribution of Experimental Data, ORNL-5689, Oak Ridge National Laboratory, Oak Ridge, Tenn., 1981.

D. E. Fields, C. W. Miller, S. E. Kurtz, and S. J. Cotter, "The Objective Validation of Model Performance; Application to Atmospheric Transport Models," EOS 62 (45) presented at Fall Meeting of the American Geological Union, San Francisco, Calif., December 1981.

D. E. Fields, C. A. Little, and D. L. Shaeffer, Users Manual for TEDPED, An Interactive Code for Testing the Hypothesis of Lognormally Distributed Data, ORNL/TM-6933, Oak Ridge National Laboratory, Oak Ridge, Tenn., 1980.

S. R. Glandon and D. E. Fields, An Interactive Statistical Distribution Analysis Program Utilizing Numerical and Graphical Methods, ORNL-5813, Oak Ridge National Laboratory, Oak Ridge, Tenn., 1982.

S. E. Kurtz and D. E. Fields, An Algorithm for Computing Significance Levels Using the Kolmogorov-Smirnov Statistic, Valid for Both Large and Small Samples, ORNL-5819, Oak Ridge National Laboratory, Oak Ridge, Tenn., 1983.

N. Smirnov, "Table for Estimating the Goodness of Fit of Empirical Distributions," Annals of Mathematical Statistics, 19, 279 (1948).

G. W. Westley and J. S. Watts (eds.), The Computing Technology Center Numerical Analysis Library, Report No. CTC-39, Union Carbide Corporation, Nuclear Division, Oak Ridge, Tenn., 1970.

APPENDIX A

APPENDIX A: SAMPLE RUN

Shown here is a sample run generated using a Decwriter III computer terminal. The terminal was used to generate the printer-plot graph included in this appendix. The terminal was assigned I/O device 6 and the data set to be analyzed, stored in file FOR59.DAT, was accessed using number 59.

```
 ASS TTY 6
TTY6 assigned

.EX TERPED,@SYS:DISPOP
LINK:   Loading
[LNKXCT TERPED execution]

 CODE TERPED/P, VERSION 6/83

 LINEAR OR LOG-NORMAL ANALYSIS? (3 CHAR)
LIN
 PERFORM KSL TEST?  (YES OR NO)
YES
 AND HOW ABOUT A CHI SQUARED TEST? (YES OR NO)
YES
 INPUT TITLE OF PLOT,UP TO 19 CHAR.
ENZYME LEVEL
 EXP. DATA INPUT:HAND INPUT OR FILE? (4 CHAR)
FILE
 ENTER INPUT FILE NUMBER
43
 ENTER NUMBER OF DATA POINTS TO BE USED.
 (IF NUMBER = 0, ALL POINTS USED FOR FILE INPUT)
0
 DATA INPUT COMPLETE; ANALYSIS STARTED.
 NUMBER OF POINTS =   21
```

```
 PRINT ORDERED DATA (YES OR NO)?
YES
   J      VALUE       PROB
   1   0.240E-04     0.0294
   2   0.780E-04     0.0765
   3   0.920E-04     0.1235
   4   0.940E-04     0.1706
   5   0.130E-03     0.2176
   6   0.180E-03     0.2647
   7   0.210E-03     0.3118
   8   0.270E-03     0.3588
   9   0.290E-03     0.4059
  10   0.410E-03     0.4529
  11   0.430E-03     0.5000
  12   0.460E-03     0.5471
  13   0.480E-03     0.5941
  14   0.770E-03     0.6412
  15   0.780E-03     0.6882
  16   0.970E-03     0.7353
  17   0.990E-03     0.7824
  18   0.110E-02     0.8294
  19   0.210E-02     0.8765
  20   0.300E-02     0.9235
  21   0.390E-02     0.9706
 DATA HAS BEEN RANK ORDERED


 LIN NORMAL ANALYSIS:
 MEAN =   7.980E-04 AND S.D. =1.016E-03

 INPUT NO. OF CELLS FOR CHI SQUARED TEST (INTEGER)
4
 BEGIN GFIT
 CHI SQUARED STATISTIC =  8.5238
NULL HYPOTHESIS WOULD BE REJECTED AT 0.004 LEVEL
 BEGIN KSL TEST
      D =  0.2408 AT A SIGNIFICANCE LEVEL OF   0.1508


 BEGIN LEAST SQUARES DATA FIT.

 COEFFICIENTS OF FIT ARE 0.885E-03 0.798E-03
 CORRELATION BETWEEN DATA AND FIT IS R= 0.840

 ENTER H TO GENERATE PLOT DATA SET, L FOR PRINTER PLOT,
R FOR RESTART, E FOR EXIT.
L

ENTER Y-AXIS LABEL (20 CHARACTERS MAXIMUM)
I.U./LITER
```

```
ENTER NUMBER OF COLUMNS FOR PLOT (60 OR 120)
120

ENTER OUTPUT DEVICE (TTY, DSK, OR LPT)
TTY

ENTER Y-AXIS ORIGIN

BETWEEN -1E8 AND 1E8
.00002


ENTER Y-AXIS STEP SIZE (UNITS/5 LINES OF PLOT,
60 LINES OF PLOT)
.00035









 ENTER H TO GENERATE PLOT DATA SET, L FOR PRINTER PLOT,
R FOR RESTART, E FOR EXIT.
R

 CODE TERPED/P, VERSION 6/83


 LINEAR OR LOG-NORMAL ANALYSIS? (3 CHAR)
LOG
 PERFORM KSL TEST?  (YES OR NO)
YES
 AND HOW ABOUT A CHI SQUARED TEST? (YES OR NO)
NO
 INPUT TITLE OF PLOT,UP TO 19 CHAR.
ENZYME
 EXP. DATA INPUT:HAND INPUT OR FILE? (4 CHAR)
FILE
 ENTER INPUT FILE NUMBER
43
 ENTER NUMBER OF DATA POINTS TO BE USED.
 (IF NUMBER = 0, ALL POINTS USED FOR FILE INPUT)
0
 DATA INPUT COMPLETE; ANALYSIS STARTED.
 NUMBER OF POINTS =   21
```

```
 PRINT ORDERED DATA (YES OR NO)?
YES
   J      VALUE       PROB
   1 -0.106E+02     0.0294
   2 -0.946E+01     0.0765
   3 -0.929E+01     0.1235
   4 -0.927E+01     0.1706
   5 -0.895E+01     0.2176
   6 -0.862E+01     0.2647
   7 -0.847E+01     0.3118
   8 -0.822E+01     0.3588
   9 -0.815E+01     0.4059
  10 -0.780E+01     0.4529
  11 -0.775E+01     0.5000
  12 -0.768E+01     0.5471
  13 -0.764E+01     0.5941
  14 -0.717E+01     0.6412
  15 -0.716E+01     0.6882
  16 -0.694E+01     0.7353
  17 -0.692E+01     0.7824
  18 -0.681E+01     0.8294
  19 -0.617E+01     0.8765
  20 -0.581E+01     0.9235
  21 -0.555E+01     0.9706
 DATA HAS BEEN RANK ORDERED


 LOG NORMAL ANALYSIS:
 MEAN =  -7.831E+00 AND S.D. =1.284E+00
 MOST PROBABLE VALUE =            7.636E-05
 MEDIAN VALUE =                   3.971E-04
 MEAN VALUE =                     9.056E-04
 99% QUANTILE =                   7.871E-03
 BEGIN KSL TEST
      D =  0.1040 AT A SIGNIFICANCE LEVEL OF   0.9355


 BEGIN LEAST SQUARES DATA FIT.

 COEFFICIENTS OF FIT ARE 0.132E+01-0.783E+01
 CORRELATION BETWEEN DATA AND FIT IS R= 0.980

 ENTER H TO GENERATE PLOT DATA SET, L FOR PRINTER PLOT,
R FOR RESTART, E FOR EXIT.
E

STOP

END OF EXECUTION
CPU TIME: 3.27  ELAPSED TIME: 11:3.32
EXIT
```

APPENDIX B

APPENDIX B: INPUT DATA USED FOR SAMPLE RUN

These data were accessed as file FOR43.DAT.

.000270	.000970
.000460	.000094
.000410	.000430
.000092	.000130
.002100	.001100
.003900	.000780
.003000	.000210
.000770	.000290
.000024	.000990
.000480	.000078

APPENDIX C

APPENDIX C: FORTRAN-10 LISTING OF THE TERPED/P CODE.

```
C TERPED/P
C CODE, TERPED BY D.E. FIELDS, C.A. LITTLE AND D.L. SHAEFFER
C CODE, TERPED, A MODIFICATION BY D.E. FIELDS
C CODE, TERPED/1, A MODIFICATION BY S.F.GLANDON AND S.E. KURTZ
C CODE MODIFIED 6/83
C        ANALYZES AND PLOTS NORMALLY OR LOG-NORMALLY DISTRIBUTED DATA
C                 AND RESULTS OF LEAST SQUARES DATA FIT ON DATA
C        ALSO PROVIDES STATISTICAL PARAMETERS DESCRIBING EXPERIMENTAL DATA
      DIMENSION DATA(300),IORD(300),PARM(2,300),B(2),TITLE(5),ICOR(2),
     1 Y(300),Y1(300)
      DATA DOL/'$    '/,E/'[illegible]'/,[illegible]/'[illegible]'/,[illegible]/'[illegible]'/,IPL/0/,AL/'L'/
      CALL COMPRS
   10 DO 20 I=1,5
   20    TITLE(I)=DOL
      WRITE(6,1000)
C        SPECIFY INPUT DATA
      CALL DSIN (DATA,ITYPE,NP,TITLE,IV,ICOR)
      IF (ITYPE.EQ.1) GO TO 30
      DO 30 I=1,NP
         DATA(I)=ALOG(DATA(I))
   30    CONTINUE
C        ORDER DATA
      CALL DSORD(DATA,NP,IORD)
C        COMPUTE PROBABILITIES AND A-VALUE
      CALL DSPROB(DATA,ITYPE,NP,IORD,PARM)
      WRITE(6,1100)
C        COMPUTE AND OUTPUT MEAN AND STANDARD DEVIATION
      CALL DSANAL(DATA,ITYPE,NP,ICOR)
C        PERFORM LEAST SQUARES FIT ON DATA
      CALL DSFIT(PARM,NP,B)
C        COMPUTE CORRELATION COEFF.
      DO 40 N=1,NP
         Y(N)=PARM(1,N)
         Y1(N)=B(1)*PARM(2,N)+B(2)
```

```
            IF (ITYPE.EQ.1) GO TO 40
            Y(N)=EXP(Y(N))
            Y1(N)=EXP(Y1(N))
   40       CONTINUE
         CALL DECOR(Y,Y1,NT,CCR)
         WRITE(6,10200) CCR
   50 WRITE(6,10300)
         READ(6,10400)A
         IF (A.EQ.T) GO TO 70
         IF (A.EQ.AL) GO TO 60
         IF (A.EQ. ) GO TO 10
         IF (A.NE.P) GO TO 50
C          PLOT DATA AND RESULTS OF FIT
         IF (IPL.EQ.0) OPEN(UNIT=1,DEVICE='DSK',FILE='SUM.FIT')
         IPL=IPL+1
         CALL DSPLOT(B,NP,PARM,ITYPE,IV,TITLE,CCR,IPL)
         GO TO 50
   60 CALL FUNCON(PARM,B,NP,ITYPE,TITLE)
         GO TO 5
   70 IF (IPL.EQ.0) GO TO 80
         CALL SETDEV(6,6)
         CALL DONPL
         CLOSE(UNIT=1,DISPOSE='LIST')
   80 STOP
10000 FORMAT(2X,/,2X,'CODE TERPED/E, VERSION 6/83',//)
10100 FORMAT(2X,'DATA HAS BEEN RANK ORDERED')
10200 FORMAT(2X,'CORRELATION BETWEEN DATA AND FIT IS P=',F6.3,/)
10300 FORMAT(2X,'ENTER: A TO GENERATE PLOT DATA SET, L FOR PRINT'
        1 'ER PLOT,',/,' R FOR RESTART, T FOR EXIT.')
10400 FORMAT(A1)
         END
CBLOCK
           BLOCK DATA
C          D.E. FIELDS, C.A. LITTLE, AND L.L. SHAEFFER
CODE, TERPED, A MODIFICATION BY D.E. FIELDS
C          DATA USED BY DATA ANALYSIS PACKAGE
         COMMON AP(2,100),DSUM,TMS,XKS(15,80)
           DATA AP/-5.[illegible], 0.[illegible]28,-4.9[illegible], 0.00000048,
```

```
&-4.80000000, 0.00000079,-4.70000000, 0.00000130,
&-4.60000000, 0.00000212,-4.50000000, 0.00000340,
&-4.40000000, 0.00000541,-4.30000000, 0.00000854,
&-4.20000000, 0.00001334,-4.10000000, 0.00002066,
&-4.00000000, 0.00003167,-3.90000000, 0.00004809,
&-3.80000000, 0.00007235,-3.70000000, 0.00010780,
&-3.60000000, 0.00015911,-3.50000000, 0.00023263,
&-3.40000000, 0.00033693,-3.30000000, 0.00048342,
&-3.20000000, 0.00068714,-3.10000000, 0.00096761,
&-3.00000000, 0.00134990,-2.90000000, 0.00186581,
&-2.80000000, 0.00255513,-2.70000000, 0.00346697,
&-2.60000000, 0.00466119,-2.50000000, 0.00620966,
&-2.40000000, 0.00819754,-2.30000000, 0.01072411,
&-2.20000000, 0.01390345,-2.10000000, 0.01786442,
&-2.00000000, 0.02275013,-1.90000000, 0.02871656,
&-1.80000000, 0.03593032,-1.70000000, 0.04456546,
&-1.60000000, 0.05479930,-1.50000000, 0.06680720,
&-1.40000000, 0.08075666,-1.30000000, 0.09680048,
&-1.20000000, 0.11506967,-1.10000000, 0.13566606,
&-1.00000010, 0.15865525,-0.90000004, 0.18406012,
&-0.80000001, 0.21185540,-0.70000005, 0.24196364,
&-0.60000002, 0.27425312,-0.50000006, 0.30853752,
&-0.40000004, 0.34457825,-0.30000001, 0.38208858,
&-0.20000005, 0.42074028,-0.10000002, 0.46017216,
&-0.00000046, 0.49999995, 0.09999996, 0.53982782,
& 0.19999993, 0.57925967, 0.29999995, 0.61791141,
& 0.39999998, 0.65542173, 0.49999994, 0.69146244,
& 0.59999996, 0.72574686, 0.69999993, 0.75803632,
& 0.79999995, 0.78814459, 0.89999998, 0.81593987,
& 0.99999994, 0.84134473, 1.10000000, 0.86433393,
& 1.19999990, 0.88493031, 1.30000000, 0.90319951,
& 1.40000000, 0.91924334, 1.49999990, 0.93319279,
& 1.60000000, 0.94520070, 1.69999990, 0.95543453,
& 1.80000000, 0.96406968, 1.90000000, 0.97128344,
& 1.99999990, 0.97724986, 2.10000000, 0.98213558,
& 2.19999990, 0.98609655, 2.30000000, 0.98927589,
& 2.40000000, 0.99180246, 2.49999990, 0.99379034,
& 2.60000000, 0.99533881, 2.69999990, 0.99653303,
```

```
     &  2.8[illegible], 0.99744487, 2.900[illegible], 0.99813419,
     &  3.[illegible], 0.99865010, 3.[illegible], 0.99903240,
     &  3.19999999, 0.99931286, 3.30000000, 0.99951658,
     &  3.4[illegible], 0.99966397, 3.50000000, 0.99976737,
     &  3.59999999, 0.99984089, 3.69999999, 0.99989221,
     &  3.8[illegible], 0.99992765, 3.90000000, 0.99995191,
     &  4.00000000, 0.99996833, 4.09999999, 0.99997935,
     &  4.19999999, 0.99998666, 4.30000000, 0.99999146,
     &  4.40000000, 0.99999459, 4.50000000, 0.99999667,
     &  4.59999999, 0.99999789, 4.69999999, 0.99999871,
     &  4.80000000, 0.99999921, 4.90000000, 0.99999952,
     &  4.99999999, 0.99999972/
      DATA XKS/ 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
     &  1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
     &  1.00000, 1.00000,
     &  .50000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
     &  1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
     &  1.00000,
     &  .22222, .92593, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
     &  1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
     &  1.00000,
     &  .09375, .31250, .79219, 1.00000, 1.00000, 1.00000, 1.00000,
     &  1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
     &  1.00000,
     &  .03840, .26912, .76992, .99936, 1.00000, 1.00000, 1.00000,
     &  1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
     &  1.00000,
     &  .01543, .57656, .93441, .99623, .99996, 1.00000, 1.00000,
     &  1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
     &  1.00000,
     &  .00612, .47446, .88937, .98911, .99960, 1.00000, 1.00000,
     &  1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
     &  1.00000,
     &  .00245, .38659, .83842, .97741, .99845, .99996, 1.00000,
     &  1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
     &  1.00000,
     &  .00094, .31261, .78442, .96121, .99615, .99982, 1.00000,
     &  1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
```

```
     & 1.00000,
     & .00036, .25128, .72046, .94131, .99222, .99943, .99998, 1.00000,
     & 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
     & .00014, .20100, .67582, .91747, .98648, .99865, .99993, 1.00000,
     & 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
     & .00005, .16014, .62209, .89126, .97885, .99732, .99979, .99999,
     & 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
     & .00002, .12715, .57136, .86304, .96935, .99530, .99953, .99997,
     & 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
     & .00001, .10066, .52323, .83337, .95807, .99250, .99908, .99993,
     & 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
     & [illegible], .07951, .47795, .80275, .94517, .98882, .99837, .99984,
     & .99990, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
     & [illegible], .06205, .43564, .77158, .93081, .98425, .99736, .99968,
     & .99997, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
     & [illegible], .04827, .3963[illegible], .74019, .91517, .97875, .99598, .99944,
     & .99994, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
     & [illegible], .03869, .35991, .70887, .89844, .97235, .99419, .99907,
     & .99989, .99999, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
     & [illegible], .03133, .32636, .67784, .88079, .96506, .99195, .99856,
     & .99981, .99998, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
     & [illegible], .02374, .29553, .64728, .86237, .95693, .98924, .99788,
     & .99968, .99996, 1.00000, 1.00000, 1.00000, 1.00000, 1.00000,
     & [illegible], .01857, .26729, .61733, .84335, .94802, .98605, .99700,
     & .99949, .99993, .99999, 1.00000, 1.00000, 1.00000, 1.00000,
     & [illegible], .01450, .24147, .58811, .82380, .93837, .98236, .99590,
     & .99924, .99989, .99999, 1.00000, 1.00000, 1.00000, 1.00000,
     & [illegible], .01132, .21793, .55970, .80401, .92805, .97817, .99456,
     & .99890, .99982, .99998, 1.00000, 1.00000, 1.00000, 1.00000,
     & .00000, .00882, .19650, .53216, .78392, .91712, .97349, .99296,
     & .99846, .99973, .99996, 1.00000, 1.00000, 1.00000, 1.00000,
     & .00000, .00687, .17702, .50554, .76308, .90565, .96832, .99110,
     & .99792, .99960, .99994, .99999, 1.00000, 1.00000, 1.00000,
     & .00000, .00535, .15935, .47987, .74338, .89303, .96269, .98895,
     & .99725, .99943, .99990, .99999, 1.00000, 1.00000, 1.00000,
     & .00000, .00416, .14334, .45517, .72309, .88129, .95661, .98651,
     & .99645, .99921, .99985, .99998, 1.00000, 1.00000, 1.00000,
     & .00000, .00323, .12885, .43145, .70288, .86851, .95010, .98378,
```

```
(187)    8 ,99551, ,99894, ,99979, ,99997, 1.00000, 1.00000, 1.00000,
(188)    8 ,00000, ,00251, ,11575, ,40870, ,68280, ,85541, ,94318, ,98076,
(189)    8 ,99441, ,99861, ,99971, ,99995, ,99999, 1.00000, 1.00000,
(190)    8 ,0000 , ,00195, ,10392, ,38693, ,66294, ,84203, ,93588, ,97745,
(191)    8 ,99315, ,99821, ,9996 , ,99992, ,99999, 1.00000, 1.00000,
(192)    8 ,00000, ,00151, ,09325, ,36612, ,64323, ,82843, ,92822, ,97384,
(193)    8 ,99172, ,99773, ,99946, ,99989, ,99998, 1.00000, 1.00000,
(194)    8 ,0000 , ,00117, ,08363, ,34624, ,62382, ,81463, ,92022, ,96995,
(195)    8 ,99312, ,99717, ,99934, ,99985, ,99997, 1.00000, 1.00000,
(196)    8 ,00000, ,00101, ,07497, ,32720, ,60470, ,80169, ,91192, ,96578,
(197)    8 ,98834, ,99652, ,99911, ,99980, ,99996, 1.00000, 1.00000,
(198)    8 ,00000, ,00071, ,06717, ,30923, ,58590, ,78663, ,90332, ,96134,
(199)    8 ,98638, ,99578, ,99886, ,99973, ,99994, ,99999, 1.00000,
(200)    8 ,00000, ,00054, ,06016, ,29225, ,56744, ,77250, ,89447, ,95664,
(201)    8 ,98423, ,99494, ,99857, ,99965, ,99992, ,99999, 1.00000,
(202)    8 ,00000, ,00042, ,05386, ,27570, ,54934, ,75831, ,88538, ,95168,
(203)    8 ,98191, ,99389, ,99824, ,99954, ,99991, ,99998, 1.00000,
(204)    8 ,00000, ,00033, ,04827, ,26019, ,53161, ,74410, ,87618, ,94648,
(205)    8 ,97939, ,99294, ,99789, ,99942, ,99986, ,99997, ,99999,
(206)    8 ,00000, ,00025, ,04312, ,24544, ,51427, ,72990, ,86658, ,94104,
(207)    8 ,97674, ,99178, ,99741, ,99928, ,99982, ,99996, ,99999,
(208)    8 ,00000, ,00021, ,03850, ,23145, ,49733, ,71572, ,85630, ,93539,
(209)    8 ,97392, ,99054, ,99692, ,99911, ,99977, ,99995, ,99999,
(210)    8 ,00000, ,00015, ,03448, ,21819, ,48078, ,70159, ,84707, ,92952,
(211)    8 ,97177, ,98911, ,99636, ,99891, ,99971, ,99993, ,99999,
(212)    8 ,00000, ,00012, ,03081, ,20562, ,46464, ,68752, ,83711, ,92345,
(213)    8 ,96754, ,98750, ,99573, ,99868, ,99963, ,99991, ,99998,
(214)    8 ,00000, ,00009, ,02753, ,19373, ,44891, ,67354, ,82702, ,91719,
(215)    8 ,96413, ,98595, ,99504, ,99842, ,99955, ,99988, ,99997,
(216)    8 ,00000, ,00007, ,02459, ,18247, ,43359, ,65965, ,81684, ,91075,
(217)    8 ,96056, ,98431, ,99428, ,99813, ,99945, ,99985, ,99996,
(218)    8 ,00000, ,00005, ,02195, ,17181, ,41862, ,64588, ,80657, ,90415,
(219)    8 ,95683, ,98255, ,99344, ,99779, ,99933, ,99982, ,99995,
(220)    8 ,00000, ,00004, ,01960, ,16174, ,40416, ,63223, ,79623, ,89739,
(221)    8 ,95299, ,98068, ,99253, ,99742, ,99919, ,99977, ,99994,
(222)    8 ,00000, ,00003, ,01750, ,15222, ,39018, ,61872, ,78583, ,89048,
(223)    8 ,94933, ,97922, ,99154, ,99701, ,99904, ,99972, ,99993,
```

```
& .00000, .00002, .01561, .14323, .37639, .60530, .77539, .88344,
& .94467, .97598, .99047, .99655, .99886, .99966, .99991,
& .00000, .00002, .01393, .13474, .36310, .59215, .76492, .87628,
& .94133, .97363, .98933, .99605, .99860, .99959, .99988,
& .00000, .00001, .01242, .12672, .35120, .57911, .75442, .86899,
& .93584, .97115, .98810, .99550, .99844, .99951, .99986,
& .00000, .00001, .01118, .11916, .33769, .56623, .74392, .86160,
& .93122, .96856, .98679, .99490, .99821, .99941, .99983,
& .00000, .00001, .00983, .11203, .32556, .55353, .73342, .85412,
& .92648, .96586, .98541, .99425, .99792, .99931, .99979,
& .00000, .00001, .00890, .10530, .31381, .54101, .72294, .84654,
& .92161, .96304, .98392, .99356, .99762, .99919, .99975,
& .00000, .00001, .00785, .09896, .30242, .52808, .71247, .83889,
& .91662, .96011, .98237, .99280, .99729, .99906, .99970,
& .00000, .00000, .00699, .09298, .29140, .51654, .70203, .83116,
& .91152, .95708, .98073, .99200, .99693, .99891, .99964,
& .00000, .00000, .00623, .08735, .28073, .50459, .69162, .82337,
& .90632, .95392, .97910, .99113, .99654, .99875, .99958,
& .00000, .00000, .00555, .08205, .27041, .49283, .68126, .81552,
& .90112, .95069, .97720, .99022, .99611, .99857, .99951,
& .00000, .00000, .00494, .07706, .26042, .48128, .67094, .80762,
& .89562, .94734, .97531, .98924, .99565, .99837, .99943,
& .00000, .00000, .00440, .07236, .25077, .46992, .66068, .79968,
& .89013, .94397, .97334, .98821, .99515, .99815, .99934,
& .00000, .00000, .00392, .06793, .24144, .45876, .65049, .79171,
& .88455, .94036, .97129, .98712, .99462, .99791, .99925,
& .00000, .00000, .00349, .06377, .23242, .44780, .64035, .78370,
& .87889, .93674, .96916, .98598, .99406, .99765, .99914,
& .00000, .00000, .00310, .05986, .22371, .43705, .63029, .77567,
& .87316, .93302, .96695, .98477, .99345, .99737, .99902,
& .00000, .00000, .00276, .05617, .21529, .42649, .62030, .76761,
& .86736, .92921, .96466, .98351, .99281, .99707, .99889,
& .00000, .00000, .00246, .05271, .20717, .41614, .61040, .75955,
& .86151, .92533, .96230, .98218, .99212, .99674, .99874,
& .00000, .00000, .00219, .04946, .19933, .40599, .60057, .75148,
& .85557, .92136, .95986, .98080, .99140, .99639, .99858,
& .00000, .00000, .00195, .04640, .19176, .39603, .59083, .74340,
```

```
      &  .84956,  .91731,  .95735,  .97936,  .99063,  .99602,  .99841,
      &  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,
      &  .84355,  .91329,  .95476,  .97786,  .98983,  .99562,  .99823,
      &  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,
      &  .83746,  .90901,  .95211,  .97630,  .98898,  .99519,  .99803,
      &  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,
      &  .83123,  .90475,  .94938,  .97469,  .98809,  .99474,  .99781,
      &  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,
      &  .82516,  .90042,  .94659,  .97291,  .98716,  .99425,  .99758,
      &  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,
      &  .81895,  .89614,  .94373,  .97128,  .98619,  .99374,  .99733,
      &  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,
      &  .81271,  .89159,  .94081,  .96950,  .98518,  .99321,  .99707,
      &  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,
      &  .80644,  .88719,  .93781,  .96765,  .98412,  .99264,  .99678,
      &  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,
      &  .80014,  .88253,  .93476,  .96576,  .98302,  .99204,  .99648,
      &  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,
      &  .79382,  .87792,  .93165,  .96380,  .98187,  .99142,  .99616,
      &  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,
      &  .78746,  .87326,  .92848,  .96180,  .98069,  .99076,  .99582,
      &  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,
      &  .78112,  .86856,  .92525,  .95974,  .97946,  .99008,  .99546,
      &  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,
      &  .77475,  .86381,  .92197,  .95762,  .97819,  .98936,  .99508,
      &  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,
      &  .76836,  .85902,  .91864,  .95546,  .97687,  .98861,  .99468,
      &  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,
      &  .76197,  .85419,  .91525,  .95324,  .97552,  .98783,  .99426,
      &  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,  .     ,
      &  .75557,  .84935,  .91182,  .95098,  .97412,  .98702,  .99382/
      END
      SUBROUTINE ALSQ(X,Y,N,Z,NN,MM,A)                                  ALSQ   0
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               ALSQ   5
      DIMENSION X(NN,3),Y(300),A(2)                                     ALSQ  10
      N = NN                                                            ALSQ  15
      N1 = N+1                                                          ALSQ  20
```

```
      M = MM                                                            ALSQ  25
      M1 = M+1                                                          ALSQ  30
      MM1 = M-1                                                         ALSQ  35
C                                                                       ALSQ  40
C     REDUCE THE LEAST SQUARES MATRIX TO UPPER TRIANGULAR FORM          ALSQ  45
C                                                                       ALSQ  50
      DO 60 L=1,M                                                       ALSQ  55
         SS = 0.                                                        ALSQ  60
         DO 10 I=L,N                                                    ALSQ  65
   10       SS = SS + A(I,L)**2                                        ALSQ  70
         S2 = SS                                                        ALSQ  75
         S =SQRT(S2)                                                    ALSQ  80
         IF (A(L,L).LT.0.) S=-S                                         ALSQ  85
         D = S2 + S*A(L,L)                                              ALSQ  90
         A(L,L) = A(L,L) + S                                            ALSQ  95
         IF (L.EQ.M) GO TO 50                                           ALSQ 100
         L1 = L+1                                                       ALSQ 105
         DO 30 J=L1,M                                                   ALSQ 110
            PE = 0.                                                     ALSQ 115
            DO 20 I=L,N                                                 ALSQ 120
   20          PE = PE + A(I,L)*A(I,J)                                  ALSQ 125
   30       A(M1,J) = PE/D                                              ALSQ 130
         DO 40 J=L1,M                                                   ALSQ 135
            DO 40 I=L,N                                                 ALSQ 140
   40          A(I,J) = A(I,J) - A(I,L)*A(M1,J)                         ALSQ 145
   50    A(M1,L) = -S                                                   ALSQ 150
   60    CONTINUE                                                       ALSQ 155
      GO TO 71                                                          ALSQ 160
C                                                                       ALSQ 165
C     REDUCE THE VECTOR Y                                               ALSQ 170
C                                                                       ALSQ 175

      M = MMM                                                           ALSQ 180
      MM1 = M-1                                                         ALSQ 185
   70 DO 80 I=1,N                                                       ALSQ 190
   80    A(I,M1) = Y(I)                                                 ALSQ 195
      DO 100 I=1,M                                                      ALSQ 200
```

```
      PP = 0.
      DO 90 I=L,N
   90    PP = PP + A(I,L)*A(I,M1)
      D = PP/(-A(L,L)*A(N1,L))
      DO 100 I=L,N
  100    A(I,M1) = A(I,M1) - D*A(I,L)
C
C     CALCULATE THE COEFFICIENT VECTOR B
C
      B(M) = A(M,M1)/A(N1,M)
      IF (M.EQ.1) GO TO 130
      DO 120 LL=1,MM1
         L = M-LL
         L1 = L+1
         PP = A(L,M1)
         DO 110 I=L1,M
  110       PP = PP - A(L,I)*B(I)
  120    B(L) = PP/A(N1,L)
C
C     CALCULATE R2
C
  130 SS = 0.
      MP1 = M+1
      DO 140 I=MP1,N
         SS = SS + A(I,M1)**2
  140    A(I,M1) = 0.
      R2 = SS
C
C     PERFORM THE BACK CALCULATIONS
C
      DO 170 LL=1,M
         L = M-LL+1
         PP = 0.
         DO 150 I=L,N
  150       PP = PP + A(I,L)*A(I,M1)
         D = PP/(-A(L,L)*A(N1,L))
         DO 160 I=L,N
```

```
  160      A(I,M1) = A(I,M1) - D*A(I,L)
  170    CONTINUE
      RETURN
      END
      SUBROUTINE DSANAL(DATA,ITYPE,NP,ICON)
C       D.E. FIELDS, C.A. LITTLE, AND D.L.SHAEFFER
CODE, TESTED, & MODIFICATION BY D.E. FIELDS
      COMMON A2(2,1 1),DSUM,RMS,XKS(15,80)
      EXTERNAL DSNORM
      DIMENSION DATA(300),ICON(2),CELLS(10),COMP(10),PDIF(6)
      REAL MEDIAN, MOST,MEAN
      DSUM=0.0
      RMS=0.0
      IF (ITYPE.EQ.2) WRITE(6,10000)
      IF (ITYPE.EQ.1) WRITE(6,10100)
C       MEAN AND STANDARD DEVIATION
      DO 10 I=1,NP
   10    DSUM=DSUM+DATA(I)
      DSUM=DSUM/NP
      DO 20 I=1,NP
   20    RMS=RMS+(DSUM-DATA(I))**2
      RMS=SQRT(RMS/(NP-1))
      SIGSQ=RMS*RMS
      WRITE(6,10200)DSUM,RMS
      IF (ITYPE.NE.2) GO TO 30
      MEDIAN=EXP(DSUM)
      MEAN=MEDIAN*EXP(SIGSQ/2.)
      MOST=MEDIAN*EXP(-SIGSQ)
      X99=MEDIAN*EXP(2.326*RMS)
      WRITE(6,10300) MOST
      WRITE(6,10400) MEDIAN
      WRITE(6,10500) MEAN
      WRITE(6,10600) X99
   30 CONTINUE
      IF (ICON(2).NE.1) GO TO 40
      WRITE(6,10700)
      READ(6,10800) K
```

```
      IDF=2                                                              DSAN 165
      CALL GFIT(DSNORM,K,DATA,NP,CELLS,COMP,CS,IDF,Q,IER)                DSAN 170
      WRITE(6,10900)CS                                                   DSAN 175
      WRITE(6,11000)Q                                                    DSAN 180
   40 CONTINUE                                                           DSAN 185
      IF (ICOS(1).NE.1) GO TO 50                                         DSAN 190
      WRITE(6,11100)                                                     DSAN 195
      CALL KNS(DSNORM,DATA,NP,PDIF(4),PDIF(6))
      IF (IER.EQ.33) WRITE(6,11200)                                      DSAN 205
      WRITE(6,11300)PDIF(4),PDIF(6)                                      DSAN 210
      WRITE(6,11400)                                                     DSAN 215
   50 CONTINUE                                                           DSAN 220
      RETURN                                                             DSAN 225
10000 FORMAT(/,2X,'LOG NORMAL ANALYSIS:')                                DSAN 230
10100 FORMAT(/,2X,'LIN NORMAL ANALYSIS:')                                DSAN 235
10200 FORMAT(2X,'MEAN = ',1PE11.3,' AND S.D. =',E9.3)                    DSAN 240
10300 FORMAT('  MOST PROBABLE VALUE =',T30,1PE14.3)                      DSAN 245
10400 FORMAT('  MEDIAN VALUE =',T30,1PE14.3)                             DSAN 250
10500 FORMAT('  MEAN VALUE =',T30,1PE14.3)                               DSAN 255
10600 FORMAT('  90% QUANTILE =',T30,1PE14.3)                             DSAN 260
10700 FORMAT(/,2X,'INPUT NO. OF CELLS FOR CHI SQUARED TEST (INTEGER)')   DSAN 265
10800 FORMAT(I2)                                                         DSAN 270
10900 FORMAT('  CHI SQUARED STATISTIC =',F8.4)                           DSAN 275
11000 FORMAT(' NULL HYPOTHESIS WOULD BE REJECTED AT',F6.3,' LEVEL')      DSAN 280
11100 FORMAT('  BEGIN KSL TEST')                                         DSAN 285
11200 FORMAT('  ASYMPTOTIC APPROXIMATION FOR N>30:')                      DSAN 290
11300 FORMAT(7X,'D =',F8.4,' AT A SIGNIFICANCE LEVEL OF ',F8.4)          DSAN 295
11400 FORMAT(/,2X)                                                       DSAN 300
      END                                                                DSAN 305
      SUBROUTINE DSCOR(X,Y,N,COR)                                        DSCO   0
C       CALLED BY DSAMAL OF DATA STATISTICS PKG.                         DSCO   5
C         BY D.E. FIELDS, C.A. LITTLE, AND D.L. SHAFFER                  DSCO  10
CODE, TESTED, & MODIFICATION BY D.E. FIELDS                              DSCO  15
      DIMENSION X(300),Y(300)                                            DSCO  20
      XSUM=0.                                                            DSCO  25
      YSUM=0.                                                            DSCO  30
      XYSUM=0.                                                           DSCO  35
      X2SUM=0.                                                           DSCO  40
      Y2SUM=0.                                                           DSCO  45
```

```
      DO 10 I=1,N                                                       DSCC  50
         XSUM = XSUM + X(I)                                             DSCC  55
         YSUM = YSUM + Y(I)                                             DSCC  60
   10    CONTINUE                                                       DSCC  65
      YBAR = YSUM / N                                                   DSCC  70
      XBAR = XSUM / N                                                   DSCC  75
      DO 20 I=1,N                                                       DSCC  80
         XYSUM = XYSUM + ((X(I)-XBAR)*(Y(I)-YBAR))                      DSCC  85
         X2SUM = X2SUM + (X(I)-XBAR)**2                                 DSCC  90
         Y2SUM = Y2SUM + (Y(I)-YBAR)**2                                 DSCC  95
   20    CONTINUE                                                       DSCC 100
      COR = XYSUM / SQRT(X2SUM*Y2SUM)                                   DSCC 105
      RETURN                                                            DSCC 110
      END                                                               DSCC 115
      SUBROUTINE DSFIT(PARM,NP,B)                                       DSFI   0
C       CALLED BY DATA STATISTICS PKG.                                  DSFI   5
C       D.E. FIELDS, C.A. LITTLE, AND D.L. SHAEFFER                     DSFI  10
CODE, TESTED, A MODIFICATION BY D.E. FIELDS                             DSFI  15
      DIMENSION PARM(2,300),A(301,3),Y(300),B(2)                        DSFI  20
C       PARM CONTAINS PAIR ORDERED DATA ELEMENTS AND 'A' VALUE          DSFI  25
C       FIT TO LINEAR FUNCTION PARM(1,N)=B(1)*PARM(2,N)+B(2)            DSFI  30
      DO 10 N=1,NP                                                      DSFI  35
         Y(N)=PARM(1,N)                                                 DSFI  40
         A(N,1)=PARM(2,N)                                               DSFI  45
         A(N,2)=1.                                                      DSFI  50
   10    CONTINUE                                                       DSFI  55
      N=NP                                                              DSFI  60
      WRITE(6,10000)                                                    DSFI  65
      CALL ALSQ(A,Y,D,B,2,N,2,301)                                      DSFI  70
      WRITE(6,10100) (B(NC),NC=1,2)                                     DSFI  75
      RETURN                                                            DSFI  80
10000 FORMAT(2X,'BEGIN LEAST SQUARES DATA FIT.',/)                      DSFI  85
10100 FORMAT(2X,'COEFFICIENTS OF FIT ARE',2E10.3)                       DSFI  90
      END                                                               DSFI  95
      SUBROUTINE DSINP(DATA,ITYPE,NP,TITLE,IV,ICOR)
C       ACCEPTS OR SPECIFIES INPUT FOR DATA STATISTICS PGM              DSIN   5
```

```
C         D.E. FIELDS, C.J. LITTLE, AND D.L. SHAEFFER
CODE, TYPED, A MODIFICATION BY D.E. FIELDS
      DIMENSION DATA (300),TITLE(5),ICOF(2)
      DATA H/'HAND'/,F/'FILE'/,LIN/'LIN'/,LOG/'LOG'/
      DATA YES/'YES'/,NC/'NO'/
      ICOF(1)=0
      ICOF(2)=0
C         SET DATA TYPE, LINEAR OR LOG-NORMAL
   10 WRITE(6,10000)
      READ(5,10100) LT
      IF (LT.NE.LIN) GO TO 20
      ITYPE=1
      GO TO 40
   20 IF (LT.EQ.LOG) GO TO 30
      WRITE(6,10200)
      GO TO 10
   30 ITYPE=2
   40 CONTINUE
C         PROVIDE FITTING TEST
      WRITE(6,10300)
   50 READ(5,10100) L
      IF (L.NE.YES) GO TO 60
      ICOF(1)=1
      GO TO 70
   60 IF (L.EQ.NO) GO TO 70
      WRITE(6,10200)
      GO TO 50
   70 WRITE(6,10400)
   80 READ(5,10100) L
      IF (L.NE.YES.AND.L.NE.NO) WRITE(6,10200)
      IF (L.NE.YES.AND.L.NE.NO) GO TO 80
      IF (L.EQ.YES) ICOF(2)=1
C         SPECIFY ORDINATE TYPE
   90 CONTINUE
      IF(LT.NE.LIN) GO TO 100
      IV=1
      GOTO 120
```

```
      100   IF(LI.T).LOG) GO TO 11
            WRITE(6,11200)
            GO TO 9
      110   IV=2
      120 CONTINUE                                                    DSIN 215
C           SPECIFY TITLE                                           DSIN 220
            WRITE(6,11500)                                          DSIN 225
            READ(6,11600) (TITLE(I),I=1,4)                          DSIN 230
C           SPECIFY INPUT DATA ORIGIN                               DSIN 235
      130   WRITE(6,11700)                                          DSIN 240
            READ(6,11800)A                                          DSIN 245
            IF (A.NE.B) GO TO 140                                   DSIN 250
            WRITE(6,11900)                                          DSIN 255
            READ(6,11000)NIN                                        DSIN 260
            GO TO 160                                               DSIN 265
      140   IF (A.EQ.H) GO TO 150                                   DSIN 270
            WRITE(6,12000)                                          DSIN 275
            GO TO 130                                               DSIN 280
      150   NIN=6                                                   DSIN 285
      160 CONTINUE                                                    DSIN 290
C           INPUT NUMBER OF POINTS                                  DSIN 295
      170   WRITE(6,11100)                                          DSIN 300
            WRITE(6,11200)                                          DSIN 305
            READ(6,11300)NP                                         DSIN 310
            IF (NP.LE.300) GO TO 190                                DSIN 315
            WRITE(6,11400)                                          DSIN 320
            GO TO 170                                               DSIN 325
      180   IF (NIN.NE.6) GO TO 190                                 DSIN 330
            WRITE(6,11500)                                          DSIN 335
            READ(NIN,11600) (DATA(I),I=1,NP)                        DSIN 340
            GO TO 210                                               DSIN 345
      190   IF (NP.NE.0) READ(NIN,11600) (DATA(I),I=1,NP)           DSIN 350
            IF (NP.NE.0) GO TO 210                                  DSIN 355
            DO 200 I=1,300                                          DSIN 360
               READ(NIN,11600,END=210)DATA(I)                       DSIN 365
               NP=I                                                 DSIN 370
      200      CONTINUE                                             DSIN 375
```

```
      210 REWIND NIN                                                    DSIN 380
          WRITE(6,11700)                                                DSIN 385
          WRITE(6,11800) NP                                             DSIN 390
          RETURN                                                        DSIN 395
11000 FORMAT(2X,'LINEAR OR LOG-NORMAL ANALYSIS? (3 CHAR)')              DSIN 400
11100 FORMAT(A3)                                                        DSIN 405
11200 FORMAT(2X,'INVALID INPUT; TRY AGAIN.')                            DSIN 410
11300 FORMAT(2X,'PERFORM KSI TEST?  (YES OR NO)')                       DSIN 415
11400 FORMAT(2X,'AND HOW ABOUT A CHI SQUARED TEST? (YES OR NO)')        DSIN 420
11500 FORMAT(2X,'INPUT TITLE OF PLOT,UP TO 19 CHAR.')                   DSIN 425
11600 FORMAT(4A5)                                                       DSIN 430
11700 FORMAT(2X,'EVP, DATA INPUT:HAND INPUT OR FILE? (4 CHAR)')         DSIN 435
11800 FORMAT(I4)                                                        DSIN 440
11900 FORMAT(2X,'ENTER INPUT FILE NUMBER')                              DSIN 445
11000 FORMAT(I2)                                                        DSIN 450
11100 FORMAT(2X,'ENTER NUMBER OF DATA POINTS TO BE USED.')              DSIN 455
11200 FORMAT(2X,'(IF NUMBER = 0, ALL POINTS USED FOR FILE INPUT)')      DSIN 460
11300 FORMAT(I3)                                                        DSIN 465
11400 FORMAT(2X,'NUMBER OF POINTS CANNOT EXCEED 300')                   DSIN 470
11500 FORMAT(2X,'ENTER DATA, ONE VALUE PER LINE')                       DSIN 475
11600 FORMAT(F12.3)                                                     DSIN 480
11700 FORMAT(2X,'DATA INPUT COMPLETE; ANALYSIS STARTED.')               DSIN 485
11800 FORMAT(2X,'NUMBER OF POINTS =',I5)                                DSIN 490
          END                                                           DSIN 495
          SUBROUTINE DSNORM(X,P)                                        DSNC   0
C  CALLED BY DATA STATISTICS PKG                                        DSNC   5
C        D.E. FIELDS, C.A. LIMIT, AND D.L. SHAFFER                      DSNC  10
C  CODE, TESTED, & MODIFICATION BY D.E. FIELDS                          DSNC  15
          COMMON AD(2,101),N,S                                          DSNC  20
          X1=(X-M)/S                                                    DSNC  25
          P=.5*(1.-ERF(-X1*.7071068))                                   DSNC  30
          RETURN                                                        DSNC  35
          END                                                           DSNC  40
          SUBROUTINE DSORD(IDST,NELS,NPTS)                              DSOR   0
C  DSORTM - MULTIPLE BYTE, LINEAR SELECTION, POINTER SORT               DSOR   5
C  PROGRAM AUTHOR:  A. E. BROOKS                                        DSOR  10
C  COMPUTER; TECHNOLOGY CENTER, UNION CARBIDE CORP., NUCLEAR DIV.,      DSOR  15
```

```
C  OAK RIDGE, TENN.                                                    DSOR  20
C  GREATLY REVISED                                                     DSOR  25
C  BY D.E. FIELDS, C.A. LITTLE, AND D.L. SHAEFFER                      DSOR  30
CODE, EDITED, & MODIFICATION BY D.E. FIELDS                           DSOR  35
C        CALLED BY DATA STATISTICS PKG                                DSOR  40
C         NUMBER OF COMPARISONS=N**2/2                                DSOR  45
C         LIST=DOUBLY DIMENSIONED ARRAY TO BE SORTED                  DSOR  50
C         NDL2=HIGHEST INDEX OF SECOND DIMENSION OF LIST              DSOR  55
C         NPTR=POINTER ARRAY , DIMENSION GE NDL2                      DSOR  60
C         LST =WORKING ARRAY , DIMENSION GE NDL2                      DSOR  65
COMMENT   NO CHECK ON INITAL ORDER                                    DSOR  70
C   **********                                                        DSOR  75
      DIMENSION NPTR(NDL2)                                            DSOR  80
      REAL*4 LIST(NDL2)                                               DSOR  85
COMMENT   SET UP POINTER ARRAY                                        DSOR  90
      DO 10 I=1,NDL2                                                  DSOR  95
   10 NPTR(I)=I                                                       DSOR 100
C   **********                                                        DSOR 105
      IF (NDL2.LE.1) RETURN                                           DSOR 110
      IUP=NDL2-1                                                      DSOR 115
      DO 40 I=1,IUP                                                   DSOR 120
         JHOLD=I                                                      DSOR 125
         JLOW=I+1                                                     DSOR 130
         DO 30 J=JLOW,NDL2                                            DSOR 135
            IF (LIST(NPTR(J)).GT.LIST(NPTR(JHOLD))) GO TO 30          DSOR 140
   20       JHOLD=J                                                   DSOR 145
   30       CONTINUE                                                  DSOR 150
         NPTRH=NPTR(I)                                                DSOR 155
         NPTR(I)=NPTR(JHOLD)                                          DSOR 160
         NPTR(JHOLD)=NPTRH                                            DSOR 165
   40    CONTINUE                                                     DSOR 170
      RETURN                                                          DSOR 175
      END                                                             DSOR 180
      SUBROUTINE DSPLOT(D,NE,FARM,II,IV,TITL,COR,IPL)
C        CALLED BY DATA STATISTICS PKG.                               DSPL   5
C        D.E. FIELDS, C.A. LITTLE, AND D.L. SHAEFFER                  DSPL  10
CODE, EDITED, & MODIFICATION BY D.E. FIELDS                           DSPL  15
```

```
      COMMON AT(2,101),DSUM,IPS,XKS(15,8))
      DIMENSION B(2),PAIM(2,300),TITL(5),X(300),Y(300),LXARAY(13)
      DIMENSION IBAK(56),Y1(300),XLB(5)
      DATA (LXARAY(I),I=1,13)/'.14','.62','2.3','6.7','16','31','50',
     1 '69','84','93','97.7','99.4','99.9'/,IES/'R= $'/, DOL/'$    '/
      CALL SETDIV(1,1)
      DO 10 J=1,5
         XLB(J)=DOL
 10      CONTINUE
      WRITE(6,10000)
      READ(5,10100) (XLB(J),J=1,5)
      WRITE(6,10200)
      IF (IV.NE.1) GO TO 20
      WRITE(6,10300)
      READ(5,10400) YCB
      WRITE(6,10500)
      READ(6,10400) YST
      GO TO 30
 20   WRITE(6,10600)
      READ(6,10400) YOR
      WRITE(6,10700)
      READ(5,10400) YCY
 30   CONTINUE
      CALL BGNPL(-IPL)
      CALL SCMPLX
      CALL PAGE(8.,9.5)
      CALL YTICKS(5)
      CALL TITLE(TITL,-100,'FRACT. DISTRIBUTION$',0, XLB,100, 8., 5.)
      IF (IV.EQ.1) CALL GRAPH(-3.,.75,YCB,YST)
      IF (IV.EQ.2) CALL YLOG (-3.,.75,YOR,YCY)
C        PLOT DATA POINTS
      DO 40 N=1,NP
         Y(N)=PAIM(1,N)
         IF (IT.EQ.2) Y(N)=EXP(Y(N))
         X(N)=PAIM(2,N)
         IF (X(N).GT.3.) X(N)=3.
         IF (X(N).LT.-3.) X(N)=-3.
```

```
 6680      40    CONTINUE                                                      DSPL 205
 6690          CALL CURVE(X,Y,NP,-1)                                         DSPL 210
 6700    C       PLOT LS FIT LINE                                            DSPL 215
 6710          DO 50 N=1,NP                                                  DSPL 220
 6720             Y1(N)=B(1)*PARM(2,N)+B(2)                                  DSPL 225
 6730             IF (IT.EQ.2) Y1(N)=EXP(Y1(N))                              DSPL 230
 6740             X(N)=PARM(2,N)                                             DSPL 235
 6750             IF (X(N).GT.3.) X(N)=3.                                    DSPL 240
 6760             IF (X(N).LT.-3.) X(N)=-3.                                  DSPL 245
 6770      50    CONTINUE                                                    DSPL 250
 6780          CALL MARKER(3)                                                DSPL 255
 6790          CALL CURVE(X,Y1,NP,0)                                         DSPL 260
 6800    C       TEMP PRINT OF LINEARIZED DATA********                       DSPL 265
 6810          DO 60 N=1,NP                                                  DSPL 270
 6820      60    CONTINUE                                                    DSPL 275
 6830    C       ***** FOR SPECIAL PLOT...*                                  DSPL 280
 6840          GO TO 70                                                      DSPL 285
 6850          CALL LINES('INPUT DATA$',IPAK,1)                              DSPL 290
 6860          CALL LINES('LS FIT$',IPAK,2)                                  DSPL 295
 6870          CALL LEGEND(IPAK,2,1.,4.)                                     DSPL 300
 6880      70  CALL XLBAXS(LXARAY,1,13,8.,'CUMULATIVE PROBABILITY (%)$', 100,0., DSPL 305
 6890         1 0.)                                                          DSPL 310
 6900          GO TO 30                                                      DSPL 315
 6910          CALL MESSAG(MPS,100,1.1,3.65)                                 DSPL 320
 6920          CALL REALNO(COF,3,1.4,3.65)                                   DSPL 325
 6930      80  CALL ENDPL(IPL)                                              DSPL 330
 6940          RETURN                                                        DSPL 335
 6950   10000  FORMAT(2X,'ENTER VERTICAL AXIS LABEL')                        DSPL 340
 6960   10100  FORMAT(5A5)                                                   DSPL 345
 6970   10200  FORMAT(2X,'DEFINED ORDINATE LENGTH IS 5 INCHES...')           DSPL 350
 6980   10300  FORMAT(2X,'ENTER Y ORIGIN.')                                  DSPL 355
 6990   10400  FORMAT(F10.0)                                                 DSPL 360
 7000   10500  FORMAT(2X,'ENTER Y AXIS STEP SIZE (UNITS/INCH).')             DSPL 365
 7010   10600  FORMAT(2X,'ENTER Y ORIGIN (NOT ZERO).')                       DSPL 370
 7020   10700  FORMAT(2X,'ENTER NUMBER OF INCHES PER LOG CYCLE.')            DSPL 375
 7030          END                                                           DSPL 380
 7040          SUBROUTINE DSPICB(DATA,IDXIE,NE,ICED,PARM)                    DSPP   0
```

```
      C        CALLED BY DATA STATISTICS PACKAGE                                DSPF   5
      C        D.E. FIELDS, C.A. LITTLE, AND D.L.SHAEFFER                       DSPF  10
      CODE, IMPROVED, A MODIFICATION BY D.E. FIELDS                             DSPF  15
            COMMON AP(2,101),DSUM,XMS,XKS(15,8)
            DIMENSION DATA(300),ICRD(300),BPAR(2,300), X(5),Y(5),BPAR(4),C(4,  DSPF  25
           1 3),P(300),A(1),B(1)                                                DSPF  30
            DATA YES/'YES'/                                                     DSPF  35
      C        ICRD SPECIFIES RANK ORDERING FOR DATA                            DSPF  40
      C        PAR1 CONTAINS DATA ELEMENTS AND A, THE RANK ORDER                DSPF  45
      C        SPLINE FIT INITIALIZATION                                        DSPF  50
            DO 10 M=1,4                                                         DSPF  55
         10    BPAR(M)=0.0                                                      DSPF  60
            NX=5                                                                DSPF  65
            IC=4                                                                DSPF  70
            IER=1                                                               DSPF  75
            M=1                                                                 DSPF  80
      C        DEFINE ELEMENTS OF PAR1                                          DSPF  85
            FNE=NP                                                              DSPF  90
            DO 50 J=1,NP                                                        DSPF  95
               JO=ICRD(J)                                                       DSPF 100
               PAR1(1,J)=DATA(JO)                                               DSPF 105
      C                                                                         DSPF 110
      C        DETERMINE 5 ADJOINING POINTS FOR SPLINE FIT                      DSPF 115
      C                                                                         DSPF 120
               FJO=JO                                                           DSPF 125
      C        COMPUTE CUMULATIVE PROBABILITY                                   DSPF 130
               P(1)=(FJO-.375)/(FNE+.25)                                        DSPF 135
               IF (P(1).GE..9999) P(1)=.9999                                    DSPF 140
               DO 20 N=1,101                                                    DSPF 145
                  IF (P(1).LT.AP(2,N)) GO TO 30                                 DSPF 150
         20       CONTINUE                                                      DSPF 155
         30    NL=N-3                                                           DSPF 160
               IF (NL.LT.1) NL=1                                                DSPF 165
               DO 40 N=1,5                                                      DSPF 170
                  NLPN=NL+N                                                     DSPF 175
                  X(N)=AP(2,NLPN)                                               DSPF 180
                  Y(N)=AP(1,NLPN)                                               DSPF 185
```

```
   40       CONTINUE
         CALL ICSICU(X,Y,NX,BPAR,C,IC,IER)
         IF (IER.GE.129) WRITE(6,10000) JC,IER
         CALL ICSEVU(X,Y,NX,C,IC,U,S,M,IER)
         IF (IER.GE.33) WRITE(6,10100)JC,IER
         R(JC)=S (1)
         PARM(2,JC)=S(1)
C
   50    CONTINUE
      WRITE(6,10200)
      READ(6,10300) YES
      IF (NES.NE.YES) GO TO 80
      WRITE(6,10400)
      DO 60 JC=1,NP
         WRITE(6,10500)JC,PARM(1,JC),R(JC)
   60    CONTINUE
      DO 70 JC=1,NP
         DATA(JC)=PARM(1,JC)
   70    CONTINUE
      RETURN
10000 FORMAT(2X,'ERROR FROM ICSICU,FROM DSPACB; DATA POINT', I2,
     1 ', ERR =',I3)
10100 FORMAT(2X,'ERROR FROM ICSEVU,FROM DSPACB; DATA POINT', I2,
     1 ', ERR =',I3)
10200 FORMAT(2X,'PRINT ORDERED DATA (YES OR NO)?')
10300 FORMAT(A3)
10400 FORMAT(4X,'J',5X,'VALUE',6X,'RPCE')
10500 FORMAT(2X,I3,E11.3,F10.4)
      END
      REAL FUNCTION ERF(X)
C16   DOUBLE PRECISION FUNCTION ERFC(X,IFAIL)
C     MARK 4 RELEASE NAG COPYRIGHT 1974.
C     EDITED BY JOYCE CLARKE OXFORD OEG NUCLEAR PHYSICS 11TH SEP 1976
C                    FORTRAN MACRO VERSION FDIA24.TEC
C     MARK 4,5 REVISED
C     ERFC(X)
      INTEGER NCF, IFAIL, J
```

```
      REAL XLO, XHI, ZERO, ONE, TWO, THREE, TWELVE, HALF, X, X2, BJP2,    ERF    40
     1 BJP1, PJ, SUM                                                      ERF    45
C                                                                         ERF    50
C87   REAL C(11)                                                         ERF    55
C87   DATA NCF/11/,XLO,XHI/-4.0,4.2/                                     ERF    60
C87   A,C(1),C(2),C(3),C(4),C(5),C(6),C(7),C(8),C(9),C(10),C(11)         ERF    65
C87   A/6.5972553E-1,-4.5366615E-1,1.6039583E-1,-4.4990818E-2,9.6414E-3  ERF    70
C87   A,-1.4425E-3,1.911E-4,1.21E-5,-3.61E-6,1.0E-7,1.0E-7/              ERF    75
C                                                                         ERF    80
      REAL C(16)                                                          ERF    85
      DATA NCF /16/, XLO, XHI /-4.0,12.0/, C(1), C(2), C(3), C(4), C(5),ERF    90
     1  C(6), C(7), C(8), C(9), C(10), C(11), C(12), C(13), C(14), C(15)ERF    95
     1   , C(16) /6.5972554958E-1,-4.5366152054E-1,1.6039834082E-1, -     ERF   100
     1 4.4990812115E-2,9.6414286E-3,-1.4425980E-3,1.0099437E-4,          ERF   105
     1 1.2085021E-5,-3.625420E-6,7.187E-8,8.914E-8,-7.42E-9,-2.39E-9,    ERF   110
     1 3.1E-10,9.E-11,-1.E-11/                                           ERF   115
C                                                                         ERF   120
C14   REAL C(21)                                                         ERF   125
C14   DATA NCF/21/,XLO,XHI/-6.0,14.0/                                    ERF   130
C14   A,C(1),C(2),C(3),C(4),C(5),C(6),C(7),C(8),C(9),C(10),C(11),C(12)  ERF   135
C14   A,C(13),C(14),C(15),C(16),C(17),C(18),C(19),C(20),C(21)           ERF   140
C14   A/6.5972554050637E-1,-4.5366152065378E-1,1.6039834082156E-1       ERF   145
C14   A,-4.4990812115112E-2,9.6414286507CE-3,-1.4425180774E-3           ERF   150
C14   A,1.0099436735E-4,1.2085)2453E-5,-3.62542060E-6,7.186811E-8       ERF   155
C14   A,8.9142363E-8,-7.42157F-9,-2.38934F-9,3.0622E-10,7.789E-11       ERF   160
C14   A,-1.097E-11,-3.09E-12,3.5E-13,1.4E-13,-1.0E-14,-1.0E-14/         ERF   165
C                                                                         ERF   170
      DATA ZERO, ONE, TWO, THREE, TWELVE, HALF /0.0,1.0,2.0,3.0,12.0,     ERF   175
     1 0.5/                                                               ERF   180
C                                                                         ERF   185
      IF (X.LT.XLO) GO TO 30                                              ERF   190
      IF (X.GE.XHI) GO TO 40                                              ERF   195
      X2 = TWO - TWELVE/(ABS(X)+THREE)                                    ERF   200
C                                                                         ERF   205
C     SUMMATION                                                           ERF   210
      BJP2 = ZERO                                                         ERF   215
      BJP1 = C(NCF)                                                       ERF   220
```

```
      J = NCP - 1                                                       EBF  225
   10 BJ = X2*BJ-1 - BJ-2 + C(J)                                        EBF  230
      IF (J.EQ.1) GO TO 20                                              EBF  235
      BJP2 = BJP1                                                       EBF  240
      BJP1 = BJ                                                         EBF  245
      J = J - 1                                                         EBF  250
      GO TO 10                                                          EBF  255
   20 SUM = HALF*(BJ-BJP2)*EXP(-XSQ)                                    EBF  260
      IF (X.LE.ZERO) GO TO 50                                           EBF  265
      PROB = SUM                                                        EBF  270
      GO TO 60                                                          EBF  275
   30 PROB = ONE                                                        EBF  280
      GO TO 60                                                          EBF  285
   40 PROB = ZERO                                                       EBF  290
      GO TO 60                                                          EBF  295
   50 PROB = ONE - SUM                                                  EBF  300
   60 RETURN                                                            EBF  305
      RETURN                                                            EBF  310
      END                                                               EBF  315
      SUBROUTINE GFIT (PDF,K,OBS,N,CELLS,COMP,CS,IDF,Q,IER)             GFIT   0
C                                                                       GFIT   5
C-GFIT----------S-------LIBRARY 1---------------------------------------GFIT  10
C                                                                       GFIT  15
C   FUNCTION              - CHI-SQUARED GOODNESS OF FIT TEST            GFIT  20
C   USAGE                 - CALL GFIT (PDF,K,OBS,N,CELLS,COMP,CS,IDF,Q,IER) GFIT  25
C   PARAMETERS   PDF      - THEORETICAL PROBABILITY DISTRIBUTION FUNCTION GFIT  30
C                             AGAINST WHICH SAMPLE IS TESTED.  PDF IS A  GFIT  35
C                             USER-SUPPLIED EXTERNAL FUNCTION.          GFIT  40
C                K        - INPUT NUMBER OF EQUIPROBABLE CELLS INTO WHICH GFIT  45
C                             THE OBSERVATIONS ARE TO BE TALLIED.       GFIT  50
C                OBS      - INPUT VECTOR OF LENGTH N, CONTAINING THE    GFIT  55
C                             SAMPLE.                                   GFIT  60
C                N        - LENGTH OF THE VECTOR OBS. (INPUT)           GFIT  65
C                CELLS    - OUTPUT VECTOR OF LENGTH K CONTAINING COUNTS GFIT  70
C                             OF OBSERVATIONS WHICH FALL INTO THE       GFIT  75
C                             EQUIPROBABLE CATEGORIES.                  GFIT  80
C                COMP     - OUTPUT VECTOR OF LENGTH K CONTAINING THE    GFIT  85
```

```
C                          COMPONENTS OF THE CHI-SQUARED STATISTIC,    GFIT  90
C                          F(CELLS-N/K)**2/N.                          GFIT  95
C                 CS     - RESULTANT CHI-SQUARED STATISTIC.           GFIT 100
C                 IDF    - ON EXIT CONTAINS THE NUMBER OF HYPOTHESIZED GFIT 105
C                          DISTRIBUTION FUNCTION PARAMETERS ESTIMATED  GFIT 110
C                          FROM THE DATA, OBS.  IT SHOULD BE ZERO      GFIT 115
C                          IF NONE WERE ESTIMATED.  ON EXIT, IT        GFIT 120
C                          CONTAINS THE DEGREES OF FREEDOM OF THE      GFIT 125
C                          STATISTIC CS.                               GFIT 130
C                 Q      - OUTPUT PROBABILITY OF THE CHI-SQUARED      GFIT 135
C                          STATISTIC EXCEEDING CS IF THE NULL          GFIT 140
C                          HYPOTHESIS IS TRUE.                         GFIT 145
C                 IER    - ERROR PARAMETER.                           GFIT 150
C                          TERMINAL ERROR = 128+N                      GFIT 155
C                            N=1 MEANS AN ERROR OCCURRED IN MDCDFI.    GFIT 160
C                            N=2 MEANS K LESS THAN 2.                  GFIT 165
C                          WARNING ERROR = 32+N                        GFIT 170
C                            N=3 MEANS EXPECTED CELL VALUE IS LESS THAN 5GFIT 175
C                            N=4 MEANS EXPECTED CELL VALUE IS LESS THAN 1GFIT 180
C                              IN THIS CASE, Q IS SET TO NEGATIVE        GFIT 185
C                              MACHINE INFINITY.                         GFIT 190
C   PRECISION              - SINGLE                                      GFIT 195
C   REQD. IMSL ROUTINES    - MDCDFI,UERTST                               GFIT 200
C   LANGUAGE               - FORTRAN                                     GFIT 205
C--------------------------------------------------------------------------GFIT 210
C   LATEST REVISION        - APRIL 3, 1975                               GFIT 215
C                                                                        GFIT 220
C                                                                        GFIT 225
      DIMENSION OBS(N),CELLS(K),COMP(K)                                  GFIT 230
      DOUBLE PRECISION CS1,Q1                                            GFIT 235
      DATA SINF1/-1.E37/                                                 GFIT 240
      WRITE(6,10000)                                                     GFIT 245
      IER = 0                                                            GFIT 250
      IF (K.LT.2) GO TO 60                                               GFIT 255
C                                     INITIALIZE COUNTS                  GFIT 260
      DO 10 I=1,K                                                        GFIT 265
   10   CELLS(I) = 0.0                                                  GFIT 270
```

```
C                                       GENERATE COUNTS                  GFIT 275
      DO 20 I=1,N                                                        GFIT 280
         X = OBS(I)                                                      GFIT 285
         CALL IDF(X,P)                                                   GFIT 290
         J = K*P + 1                                                     GFIT 295
         IF (P.EQ.1.0) J = K                                             GFIT 300
   20    CELLS(J) = CELLS(J) + 1.0                                       GFIT 305
C                                       DETERMINE EXPECTED VALUES OF CELLS GFIT 310
      EN = N                                                             GFIT 315
      EK = K                                                             GFIT 320
      E = EN/EK                                                          GFIT 325
      DO 30 I=1,K                                                        GFIT 330
         CELL1 = CELLS(I)-E                                              GFIT 335
C                                       FIND COMPONENTS OF STATISTICS    GFIT 340
   30    COMP(I) = EK*(CELL1*CELL1)/EN                                   GFIT 345
      CS1 = 0.0D0                                                        GFIT 350
      DO 40 I=1,K                                                        GFIT 355
   40    CS1 = CS1 + COMP(I)                                             GFIT 360
      CS = CS1                                                           GFIT 365
      IDF = K-1-IDF                                                      GFIT 370
      IF (E.LT.5.0) IER = 35                                             GFIT 375
      IF (E.LT.1.0) GO TO 50                                             GFIT 380
C                                       FIND PROBABILITY                 GFIT 385
      CALL MDCDFI(CS1,IDF,Q1,IET)                                        GFIT 390
      IF (IET.GT.127) GO TO 70                                           GFIT 395
      Q = Q1                                                             GFIT 400
      IF (IET-32) 90,80,80                                               GFIT 405
   50 IER = 36                                                           GFIT 410
      Q = FINPM                                                          GFIT 415
      GO TO 80                                                           GFIT 420
   60 IER = 130                                                          GFIT 425
      GO TO 80                                                           GFIT 430
   70 IER = IET                                                          GFIT 435
   80 CONTINUE                                                           GFIT 440
      CALL UERTST(IER,6HGFIT  )                                          GFIT 445
   90 RETURN                                                             GFIT 450
10000 FORMAT(2X,'BEGIN GFIT')                                           GFIT 455
```

```
      END                                                                GFIT 460
      SUBROUTINE ICSEVU (X,Y,NX,C,IC,U,S,M,IER)                          ICSE   0
C                                                                        ICSE   5
C-ICSEVU--------D-------LIBRARY 1----------------------------------------ICSE  10
C                                                                        ICSE  15
C   FUNCTION              - EVALUATION OF A CUBIC SPLINE.                ICSE  20
C   USAGE                 - CALL ICSEVU(X,Y,NX,C,IC,U,S,M,IER)           ICSE  25
C   PARAMETERS   X        - VECTOR OF LENGTH NX CONTAINING THE ABSCISSAE ICSE  30
C                             OF THE NX DATA POINTS (X(I),Y(I)) I=1,...,  ICSE  35
C                             NX (INPUT). X MUST BE ORDERED SO THAT       ICSE  40
C                             X(I) .LT. X(I+1).                           ICSE  45
C                Y        - VECTOR OF LENGTH NX CONTAINING THE ORDINATES ICSE  50
C                             (OR FUNCTION VALUES) OF THE NX DATA POINTS  ICSE  55
C                             (INPUT).                                    ICSE  60
C                NX       - NUMBER OF ELEMENTS IN X AND Y (INPUT).       ICSE  65
C                             NX MUST BE .GE. 2.                          ICSE  70
C                C        - SPLINE COEFFICIENTS (INPUT). C IS AN NX-1 BY ICSE  75
C                             3 MATRIX.                                   ICSE  80
C                IC       - ROW DIMENSION OF MATRIX C IN THE CALLING     ICSE  85
C                             PROGRAM (INPUT). IC MUST BE .GE. NX-1.      ICSE  90
C                U        - VECTOR OF LENGTH M CONTAINING THE ABCISSAE   ICSE  95
C                             OF THE M POINTS AT WHICH THE CUBIC SPLINE   ICSE 100
C                             IS TO BE EVALUATED (INPUT).                 ICSE 105
C                S        - VECTOR OF LENGTH M (OUTPUT).                 ICSE 110
C                             THE VALUE OF THE SPLINE APPROXIMATION AT    ICSE 115
C                             U(I) IS                                     ICSE 120
C                             S(I) = ((C(J,3)*D+C(J,2))*D+C(J,1))*D+Y(J)  ICSE 125
C                             WHERE X(J) .LE. U(I) .LT. X(J+1) AND        ICSE 130
C                             D = U(I)-X(J).                              ICSE 135
C                M        - NUMBER OF ELEMENTS IN U AND S (INPUT).       ICSE 140
C                IER      - ERROR PARAMETER.                             ICSE 145
C                           WARNING ERROR                                 ICSE 150
C                             IER = 33, U(I) IS LESS THAN X(1).           ICSE 155
C                             IER = 34, U(I) IS GREATER THAN X(NX).       ICSE 160
C   PRECISION             - SINGLE/DOUBLE                                ICSE 165
C   REQD. IMSL ROUTINES   - UERTST                                       ICSE 170
C   LANGUAGE              - FORTRAN                                      ICSE 175
```

```
      C-----------------------------------------------------------------------ICSE 180
      C   LATEST REVISION      - AUGUST 9, 1974                                ICSE 185
      C                                                                       ICSE 190
      C                                                                       ICSE 195
            DIMENSION X(NX),Y(NX),C(IC,3),U(M),S(M)                           ICSE 200
      C     DOUBLE PRECISION   C,D,DD,S,U,X,Y,ZERO                            ICSE 205
            I=1                                                               ICSE 210
            ZERO=0.                                                           ICSE 215
      C                                     INITIALIZE ERROR PARAMETERS       ICSE 220
            JER = 0                                                           ICSE 225
            KER = 0                                                           ICSE 230
            IF (M.LE.0) GO TO 100                                             ICSE 235
            NXM1 = NX-1                                                       ICSE 240
            IF (I.GT.NXM1) I = 1                                              ICSE 245
      C                                     EVALUATE SPLINE AT M POINTS       ICSE 250
            DO 80 K=1,M                                                       ICSE 255
      C                                     FIND THE PROPER INTERVAL          ICSE 260
               D = U(K)-X(I)                                                  ICSE 265
               IF (D) 10,50,30                                                ICSE 270
         10    IF (I.EQ.1) GO TO 60                                           ICSE 275
               I = I-1                                                        ICSE 280
               D = U(K)-X(I)                                                  ICSE 285
               IF (D) 10,50,40                                                ICSE 290
         20    I = I+1                                                        ICSE 295
               D = DD                                                         ICSE 300
         30    IF (I.GE.NX) GO TO 70                                          ICSE 305
               DD = U(K)-X(I+1)                                               ICSE 310
               IF (DD.GE.ZERO) GO TO 20                                       ICSE 315
               IF (D.EQ.ZERO) GO TO 50                                        ICSE 320
      C                                     PERFORM EVALUATION                ICSE 325
         40    S(K) = ((C(I,3)*D+C(I,2))*D+C(I,1))*D+Y(I)                     ICSE 330
               GO TO 80                                                       ICSE 335
         50    S(K) = Y(I)                                                    ICSE 340
               GO TO 80                                                       ICSE 345
      C                                     WARNING - U(I) .LT. X(1)          ICSE 350
         60    JER = 33                                                       ICSE 355
               GO TO 90                                                       ICSE 360
```

```
C                                        IF U(I) .GT. X(NX) - WARNING          ICSE 365
   70    IF (DD.GT.ZERO) KER = 34                                              ICSE 370
         D = U(K)-X(NM1)                                                       ICSE 375
         I = NX-1                                                              ICSE 380
         GO TO 40                                                              ICSE 385
   80    CONTINUE                                                              ICSE 390
      IER = MAX0(JER,KER)                                                      ICSE 395
   90 CONTINUE                                                                 ICSE 400
      IF (JER.GT.0) CALL UERTST(JER,6HICSEVU)                                  ICSE 405
      IF (KER.GT.0) CALL UERTST(KER,6HICSEVU)                                  ICSE 410
  100 RETURN                                                                   ICSE 415
      END                                                                      ICSE 420
      SUBROUTINE ICSICU (X,Y,NX,BEAR,C,IC,IER)                                 ICSI   0
C                                                                              ICSI   5
C-ICSICU--------P-------LIBRARY 1-------------------------------------------ICSI  10
C                                                                              ICSI  15
C   FUNCTION              - INDEFINITE CUBIC APPROXIMATION BY CUBIC SPLINES    ICSI  20
C                           WITH A BIRKHOFF SECOND DERIVATIVE END              ICSI  25
C                           CONDITIONS.                                        ICSI  30
C   USAGE                 - CALL ICSICU(X,Y,NX,BEAR,C,IC,IER)                  ICSI  35
C   PARAMETERS    X       - VECTOR OF LENGTH NX CONTAINING THE ABSCISSAE       ICSI  40
C                           OF THE NX DATA POINTS (X(I),Y(I)) I=1,...,         ICSI  45
C                           NX (INPUT). X MUST BE ORDERED SO THAT              ICSI  50
C                           X(I) .LT. X(I+1).                                  ICSI  55
C                 Y       - VECTOR OF LENGTH NX CONTAINING THE ORDINATES       ICSI  60
C                           (OR FUNCTION VALUES) OF THE NX DATA POINTS         ICSI  65
C                           (INPUT).                                           ICSI  70
C                 NX      - NUMBER OF ELEMENTS IN X AND Y (INPUT). NX          ICSI  75
C                           MUST BE .GE. 2.                                    ICSI  80
C                 BEAR    - VECTOR OF LENGTH 4 CONTAINING THE END            ICSI  85
C                           CONDITION PARAMETERS (INPUT).                      ICSI  90
C                           B(1)*SPP(1)+BEAR(1)*SPP(2) = BEAR(2),              ICSI  95
C                           BEAR(3)*SPP(NX-1)+BEAR(4)*SPP(NX) = BEAR(4),       ICSI 100
C                           WHERE SPP(I) = SECOND DERIVATIVE OF THE            ICSI 105
C                           CUBIC SPLINE FUNCTION S EVALUATED AT X(I).         ICSI 110
C                 C       - SPLINE COEFFICIENTS (OUTPUT). C IS AN NX-1 BY 4    ICSI 115
C                           4 MATRIX. THE VALUE OF THE SPLINE                  ICSI 120
```

```
C                                  APPROXIMATION AT T IS                    ICSI 125
C                                  S(T) = ((C(I,3)*D+C(I,2))*D+C(I,1))*D+Y(I)  ICSI 130
C                                  WHERE X(I) .LE. T .LT. X(I+1) AND        ICSI 135
C                                  D = T-X(I).                              ICSI 140
C                    IC      - LOW DIMENSION OF MATRIX C IN THE CALLING     ICSI 145
C                              PROGRAM (INPUT). IC MUST BE .GE. NX-1.       ICSI 150
C                    IER     - ERROR PARAMETER.                             ICSI 155
C                              TERMINAL ERROR                               ICSI 160
C                                IER = 129, IC IS LESS THAN NX-1.           ICSI 165
C                                IER = 130, NX IS LESS THAN 2.              ICSI 170
C                                IER = 131, INPUT ABSCISSA ARE NOT ORDERED  ICSI 175
C                                  SO THAT X(1) .LT. X(2) ... .LT. X(NX).   ICSI 180
C   PRECISION            - SINGLE/DOUBLE                                    ICSI 185
C   REQD. IMSL ROUTINES  - UERTST                                           ICSI 190
C   LANGUAGE             - FORTRAN                                          ICSI 195
C-------------------------------------------------------------------------ICSI 200
C   LATEST REVISION      - JULY 26, 1974                                    ICSI 205
C                                                                           ICSI 210
C                                                                           ICSI 215
      DIMENSION X(NX),Y(NX),BCAT(4),C(IC,3)                                 ICSI 220
C     DOUBLE PRECISION   BCAT,C,DX,DXC,DXJP1,DXT,DYJ,DYJP1,HALF,ONE,PJ,    ICSI 225
     1                   SIX,SIXI,TWO,X,Y,YPPA,YPPB,ZERO                    ICSI 230
      EQUIVALENCE (DXJ,YPPB),(PJ,SIXI),(DXJP1,YPPA)                         ICSI 235
      ZERO=0.                                                               ICSI 240
      HALF=.5                                                               ICSI 245
      ONE=1.                                                                ICSI 250
      TWO=2.                                                                ICSI 255
      SIX=6.                                                                ICSI 260
C                                     CHECK ERROR CONDITIONS                ICSI 265
      IER = 0                                                               ICSI 270
      NXM1 = NX-1                                                           ICSI 275
      IF (IC.LT.NXM1) GO TO 60                                              ICSI 280
      IF (NX.LT.2) GO TO 70                                                 ICSI 285
      IF (NX.EQ.2) GO TO 20                                                 ICSI 290
C                                     COMPUTE COEFFICIENTS AND RIGHT        ICSI 295
C                                     HAND SIDE OF THE TRIDIAGONAL          ICSI 300
C                                     SYSTEM DEFINING THE SECOND            ICSI 305
```

```
10750   C                                       DERIVATIVES OF THE SPLINE          ICSI 310
10760   C                                       INTERPOLANT FOR (X,Y)              ICSI 315
10770   C                                       C(J,1) = LAMBDA(J)                 ICSI 320
10780   C                                       C(J,2) = MU(J)                     ICSI 325
10790   C                                       C(J,3) = D(J)                      ICSI 330
10800           DXJ = X(2)-X(1)                                                    ICSI 335
10810           IF (DXJ.LE.ZERO) GO TO 80                                          ICSI 340
10820           DYJ = Y(2)-Y(1)                                                    ICSI 345
10830           DO 10 J=2,NXM1                                                     ICSI 350
10840              DXJP1 = X(J+1)-X(J)                                             ICSI 355
10850              IF (DXJP1.LE.ZERO) GO TO 80                                     ICSI 360
10860              DYJP1 = Y(J+1)-Y(J)                                             ICSI 365
10870              DXP = DXJ+DXJP1                                                 ICSI 370
10880              C(J,1) = DXJP1/DXP                                              ICSI 375
10890              C(J,2) = ONE-C(J,1)                                             ICSI 380
10900              C(J,3) = THREE*(DYJP1/DXJP1-DYJ/DXJ)/DXP                        ICSI 385
10910              DXJ = DXJP1                                                     ICSI 390
10920              DYJ = DYJP1                                                     ICSI 395
10930           10 CONTINUE                                                        ICSI 400
10940   C                                       FACTOR THE TRIDIAGONAL MATRIX      ICSI 405
10950   C                                       AND SOLVE FOR U                    ICSI 410
10960   C                                       C(J,2) = U(J)                      ICSI 415
10970   C                                       C(J,1) = Q(J)                      ICSI 420
10980   C                                       BPAR(1) = LAMBDA(1)                ICSI 425
10990   C                                       BPAR(2) = D(1)                     ICSI 430
11000   C                                       BPAR(3) = MU(NX)                   ICSI 435
11010   C                                       BPAR(4) = D(NX)                    ICSI 440
11020           20 C(1,1) = -BPAR(1)*HALF                                          ICSI 445
11030              C(1,2) = BPAR(2)*HALF                                           ICSI 450
11040              IF (NX.EQ.2) GO TO 40                                           ICSI 455
11050              DO 30 J=2,NXM1                                                  ICSI 460
11060                 PJ = C(J,2)*C(J-1,1)+TWO                                     ICSI 465
11070                 C(J,1) = -C(J,1)/PJ                                          ICSI 470
11080                 C(J,2) = (C(J,3)-C(J,2)*C(J-1,2))/PJ                         ICSI 475
11090           30    CONTINUE                                                     ICSI 480
11100   C                                       SOLVE FOR CUBIC COEFFICIENTS       ICSI 485
11110   C                                       OF SPLINE INTERPOLANT              ICSI 490
```

```
C                                         C(J,1), C(J,2), AND C(J,3)        ICSI 495
   40 YPPB = (BPAF(4)-BPAE(3)*C(NXM1,2))/(BPAF(3)*C(NXM1,1)+TWO)            ICSI 500
      SIXI = ONE/SIX                                                        ICSI 505
      DO 50 I=1,NXM1                                                        ICSI 510
         J = NX-I                                                           ICSI 515
         YPPA = C(J,1)*YPPB+C(J,2)                                          ICSI 520
         DX = X(J+1)-X(J)                                                   ICSI 525
         C(J,3) = SIXI*(YPPB-YPPA)/DX                                       ICSI 530
         C(J,2) = HALF*YPPA                                                 ICSI 535
         C(J,1) = (Y(J+1)-Y(J))/DX-(C(J,2)+C(J,3)*DX)*DX                    ICSI 540
         YPPB = YPPA                                                        ICSI 545
   50    CONTINUE                                                           ICSI 550
      GO TO 100                                                             ICSI 555
   60 IER = 129                                                             ICSI 560
      GO TO 90                                                              ICSI 565
   70 IER = 130                                                             ICSI 570
      GO TO 90                                                              ICSI 575
   80 IER = 131                                                             ICSI 580
   90 CONTINUE                                                              ICSI 585
      CALL UERTST(IER,6HICSICU)                                             ICSI 590
  100 RETURN                                                                ICSI 595
      END                                                                   ICSI 600
      SUBROUTINE LINPLT(IPLOT,YORG,YSTEP,PARM,ICOL,INP,B)                   LINP   0
      DIMENSION IPLOT(61,122),PARM(2,300),B(2)                              LINP   5
      DATA KDATA/'D'/,KFIT/'F'/,KCOF/'*'/                                   LINP  10
      YNORM=12.*YSTEP                                                       LINP  15
      XNORM=6.                                                              LINP  20
      XORG=-3.                                                              LINP  25
      DO 10 I=1,INP                                                         LINP  30
         CALL NORM(PARM(1,I),YORG,61,YNORM,IROW)                            LINP  35
         CALL NORM(PARM(2,I),XORG,ICOL,XNORM,ICOLUM)                        LINP  40
         ICOLUM=ICOLUM+1                                                    LINP  45
         IF (IROW.LT.1.OR.IROW.GT.61) GO TO 10                              LINP  50
         IF (ICOLUM.LT.2.OR.ICOLUM.GT.122) GO TO 10                         LINP  55
         IPLOT(IROW,ICOLUM)=KDATA                                           LINP  60
   10    CONTINUE                                                           LINP  65
      DO 30 M=1,60                                                          LINP  70
```

```
      YFV=((1*YNORM)/60.)+YORG                                          LINE  75
      XFV=(YFV-B(2))/B(1)                                               LINE  80
      CALL NORM(XFV,XORG,ICCL,XNORM,IFX)                                LINE  85
      IFX=IFX+1                                                         LINE  90
      IF (IFX.LT.2.OR.IFX.GT.122) GO TO 30                              LINE  95
      IF (IPLOT(M+1,IFX).EQ.KDATA) GO TO 20                             LINE 100
      IPLOT(M+1,IFX)=KFILL                                              LINE 105
      GO TO 30                                                          LINE 110
   20 IPLOT(M+1,IFX)=KCOL                                               LINE 115
   30 CONTINUE                                                          LINE 120
      RETURN                                                            LINE 125
      END                                                               LINE 130
      SUBROUTINE LOGPLT(IPLOT,YORG,ICYCLE,PARM,INP,B,ICCL)              LOGP   0
      DIMENSION IPLOT(61,122),PARM(2,300),B(2)                          LOGP   5
      DATA KDATA/'D'/,KFILL/'F'/,KCOL/'*'/                              LOGP  10
      YORG=-3.                                                          LOGP  15
      XNORM=6.                                                          LOGP  20
      IER=0                                                             LOGP  25
      M=1                                                               LOGP  30
      ITIER=60/ICYCLE                                                   LOGP  35
      ZMIN=ALOG(YORG)                                                   LOGP  40
      DO 70 L=1,ICYCLE                                                  LOGP  45
         ZMAX=ALOG(YORG)+(ALOG(10.)*L)                                  LOGP  50
         OFFAC=ZMIN                                                     LOGP  55
         IF (M.GT.INP) GO TO 40                                         LOGP  60
         DO 30 I=M,INP                                                  LOGP  65
            IF (PARM(1,I).LT.ZMIN) GO TO 30                             LOGP  70
            IF (PARM(1,I).GT.ZMAX) GO TO 40                             LOGP  75
            ALNYV=PARM(1,I)                                             LOGP  80
            CALL NORM(ALNYV,OFFAC,ITIER,ALOG(10.),IROW)                 LOGP  85
            IROW=IROW+(ITIER*(L-1))                                     LOGP  90
            IF (IROW.GE.1.AND.IROW.LE.61) GO TO 10                      LOGP  95
            IER=IER+1                                                   LOGP 100
            WRITE(6,10000)                                              LOGP 105
            IF (IER.GT.2) RETURN                                        LOGP 110
            GO TO 30                                                    LOGP 115
   10       CALL NORM(PARM(2,I),XORG,ICCL,XNORM,ICOLUM)                 LOGP 120
```

```
                ICOLUM=ICCLUM+1                                                 LOGP 125
                IF (ICOLUM.GT.2.AND.ICCLUM.LE.122) GO TO 20                     LOGP 130
                GO TO 30                                                        LOGP 135
   20           IPLOT(IROW,ICOLUM)=KDATA                                        LOGP 140
   30           CONTINUE                                                        LOGP 145
   40         M=I                                                               LOGP 150
              DO 60 K=1,ITIER                                                   LOGP 155
                 IYFV=K+(ITIER*(L-1))                                           LOGP 160
                 YFV=((IYFV*ALOG(10.))/ITIER)+ALOG(YOFG)                        LOGP 165
                 XFV=(YFV-P(2))/P(1)                                            LOGP 170
                 CALL NORM(XFV,XOFG,ICCI,XNORM,IXFV)                            LOGP 175
                 IXFV=IXFV+1                                                    LOGP 180
                 IF (IXFV.LT.2.OR.IXFV.GT.122) GO TO 60                         LOGP 185
                 IF (IPLOT(IYFV,IXFV).EQ.KDATA) GO TO 50                        LOGP 190
                 IPLOT(IYFV,IXFV)=KFIT                                          LOGP 195
                 GO TO 60                                                       LOGP 200
   50            IPLOT(IYFV,IXFV)=KCCF                                         LOGP 205
   60            CONTINUE                                                      LOGP 210
              ZMIN=ZMAX                                                         LOGP 215
   70         CONTINUE                                                         LOGP 220
              RETURN                                                            LOGP 225
 1000 FORMAT(/' ERROR IN LOG PLOT NORMALIZATION PARAMETERS')                    LOGP 230
              END                                                               LOGP 235
              SUBROUTINE MDCDFI (CS,N,Q,IER)                                    MDCD   0
C                                                                               MDCD   5
C-MDCDFI--------)-------LIBRARY 1---------------------------------------------MDCD  10
C                                                                               MDCD  15
C    FUNCTION              - CHI-SQUARED PROBABILITY DISTRIBUTION FUNCTION      MDCD  20
C    USAGE                 - CALL MDCDFI(CS,N,Q,IER)                            MDCD  25
C    PARAMETER     CS      - INPUT VALUE FROM WHICH THE INTEGRATION IS          MDCD  30
C                              PERFORMED. CS SHOULD NOT BE LESS THAN ZERO.      MDCD  35
C                  N       - INPUT DEGREES OF FREEDOM, NOT LESS THAN ONE.       MDCD  40
C                  Q       - PROBABILITY OF EXCEEDING CS.                       MDCD  45
C                  IER     - ERROR PARAMETER.                                   MDCD  50
C                              TERMINAL ERROR=128+N.                            MDCD  55
C                                N = 1 INDICATES THAT CS OR N(DF) IS OUT OF     MDCD  60
C                                RANGE.                                         MDCD  65
```

```
C   PRECISION                - DOUBLE                                          MDCD  70
C   REQD. IMSL ROUTINES      - UERTST                                          MDCD  75
C   LANGUAGE                 - FORTRAN                                         MDCD  80
C---------------------------------------------------------------------------MDCD  85
C   LATEST REVISION          - JANUARY 1, 1977                                 MDCD  90
C                                                                              MDCD  95
C                                                                              MDCD 100
      IMPLICIT REAL*8(A-H,C-Z)                                                 MDCD 105
C                                                 CHECK INPUT RANGES           MDCD 110
      IF ((CS.LT.0.D0).OR.(N.LT.1)) GO TO 80                                   MDCD 115
      P =.5D0*CS                                                               MDCD 120
      IF=1                                                                     MDCD 125
      Y=1.0                                                                    MDCD 130
      IE=1                                                                     MDCD 135
C                                                 SET EVEN DEGREES OF FREEDOM (N)  MDCD 140
C                                                 INDICATOR.                   MDCD 145
      IF (N.EQ.(N/2)*2) IE=2                                                   MDCD 150
      IF (((IE.EQ.1).OR.(N.GT.2)).AND.(A.LE.170.0D0)) Y=DEXP(-A)               MDCD 155
      S=Y                                                                      MDCD 160
C                                                 IF N IS EVEN, SET S = Q FOR IF = 2.  MDCD 165
C                                                 IF N IS ODD, SET S = Q FOR DF = 1.   MDCD 170
      IF (IE.NE.1) GO TO 10                                                    MDCD 175
      DSQA=DSQRT(A)                                                            MDCD 180
      S=0.0                                                                    MDCD 185
      IF (DSQA.LT.13.3) S = 1.-ERF(DSQA)                                       MDCD 190
   10 IF (N.GT.2) GO TO 30                                                     MDCD 195
   20 Q=S                                                                      MDCD 200
      GO TO 100                                                                MDCD 205
C                                                 USE RECURRENCE FORMULA, REFERENCE 2.  MDCD 210
   30 XD=.5D0*(N-1)                                                            MDCD 215
      Z=1.D0                                                                   MDCD 220
      IF (IE.EQ.2) Z=.5D0                                                      MDCD 225
      IF (A.LT.170.0D0) GO TO 50                                               MDCD 230
      R=1.0D                                                                   MDCD 235
      IF (IE.EQ.1) R=0.5723649429246980D0                                      MDCD 240
      C=DLOG(A)                                                                MDCD 245
   40 IF (Z.LT.XD) GO TO 20                                                    MDCD 250
```

```
12600          F=DLOG(Z)+E                                                    MDCD 255
12610          A=G=C*Z-A-E                                                    MDCD 260
12620          IF (A.GE.-17*.0D0) S=DEXP(A+G)+S                               MDCD 265
12630          Z=Z+1.0D0                                                      MDCD 270
12640          GO TO 40                                                       MDCD 275
12650       50 E=1.0D0                                                        MDCD 280
12660          C=0.0D0                                                        MDCD 285
12670          IF (A.EQ.1.0) GO TO 70                                         MDCD 290
12680          IF (IE.EQ.1) E=0.5641895835477563E0/DSQA                       MDCD 295
12690       60 E=E*A/Z                                                        MDCD 300
12700          IF (C.EQ.C+E) GO TO 70                                         MDCD 305
12710          C=C+E                                                          MDCD 310
12720          Z=Z+1.0                                                        MDCD 315
12730          IF (Z.LT.XD) GO TO 60                                          MDCD 320
12740       70 Q=C*Y+S                                                        MDCD 325
12750          GO TO 100                                                      MDCD 330
12760       80 IER=129                                                        MDCD 335
12770       90 CONTINUE                                                       MDCD 340
12780          CALL UERTST(IER,'MDCDFI')                                      MDCD 345
12790      100 RETURN                                                         MDCD 350
12800          END                                                            MDCD 355
12810
12820          SUBROUTINE NRS(BDF,X,N,DM,ALPHA)
12830    C
12840    C        THIS SUBROUTINE HAS BEEN WRITTEN WITH THE ASSUMPTION
12850    C        THAT A NORMAL DISTRIBUTION IS TO BE EVALUATED.
12860    C        IF ANOTHER TYPE OF DISTRIBUTION IS TO BE USED
12870    C        THESE LINES OF CODE MUST BE CHANGED.   THEY ARE:
12880    C
12890    C        100 XN=XN+1.25    ** REMOVE THIS LINE  **
12900    C        200 BDF1=(XJ-1.375)/XN   ** CHANGE TO **  BDF1=(XJ-1)/XN
12910    C        300 BDF2=(XI-0.375)/XN   ** CHANGE TO **  BDF2=XI/XN
12920    C
12930    C        INITIALIZE VARIABLES
12940          DIMENSION X(N)
12950          BDF1=0.
12960          BDF2=0.
```

```
      DN=0.
      XN=N
  100 XN=XN+1.25
   10 I=1
   20 J=I
C          CHECK FOR TIED DATA VALUES
   30 IF(X(I).LT.X(I+1))GOTO 40
      I=I+1
      IF(I.EQ.N)GOTO 40
      GOTO 30
   40 XI=I
      XJ=J
C          COMPUTE CUMULATIVE PROBABILITIES FROM ABOVE AND BELOW
      EDF1=(XJ-1.375)/XN
      EDF2=(XI-.375)/XN
      CALL NDF(X(I),P)
C          COMPARE FOR NEW MAXIMUM DIFFERENCE
      DN=AMAX1(DN,ABS(P-EDF1),ABS(P-EDF2))
C          INCREMENT COUNTERS AND PICK POINT OF LOOP REENTRY
C          FOR LAST DATA ELEMENT RETURN TO 40, OTHERWISE TO 20
      I=I+1
      IF(I.LT.N)GOTO 20
      J=I
      IF(I.EQ.N)GOTO 40
C          CHOOSE AND CALL PROBABILITY SUBROUTINE
      IF(N.GE.81)GOTO 50
      CALL NLE80(DN,PROB2,N)
      GOTO 60
   50 CALL NGT80(DN,PROB2,N)
   60 ALPHA=1.-PROB2
      RETURN
      END

C
      SUBROUTINE NLE80(SIAI,PROB2,N)
      COMMON AD(2,101),DSUM,RMS,XKS(15,60)
      DIFS=0.1
```

```
      XN=N
C         FIND VALUES ON EACH SIDE OF STAT, EXPRESSIBLE AS C/N,
C         WHERE C IS AN INTEGER
      DO 10 I=1,N
      XI=I
      DIF=XI/XN
      IF(STAT.LT.DIFS)GO TO 5
      IF(STAT.LT.DIF)GOTO 20
    5 DIFS=DIF
   10 ISAVE=I
   20 ISAVE1=ISAVE+1
C         ACCESS PROBABILITIES FROM BLOCK DATA
      IF(ISAVE.EQ.0)GOTO 30
      FLOW=XKS(ISAVE,N)
      FHIGH=XKS(ISAVE1,N)
      GOTO 40
   30 FLOW=0.
      FHIGH=XKS(ISAVE1,N)
C         INTERPOLATE PROBABILITIES
   40 FRAC=(STAT-DIFS)/(DIF-DIFS)
      PROB2=FLOW+(FHIGH-FLOW)*FRAC
      RETURN
      END

      SUBROUTINE NGT80(STAT,PROB2,N)
C
C     NGT80 YIELDS THE PROBABILITY FOR A 2-SIDED K-S TEST
C
C     FORM THE WEIGHTED STATISTIC Z FROM THE ORIGINAL
C     STATISTIC D AND N AND THE SAMPLE SIZE N
      XN=N
      Z=STAT*SQRT(XN)
      Z80=Z*Z
C         SELECT FORMULA FOR  AND COMPUTE PROBABILITY,
C         BASED ON RANGE OF WEIGHTED STATISTIC
C                 0.22 < Z
      IF(Z.GT.0.22) GOTO 10
```

```
      PROB2=0.
      GOTO 40
C                 0.22 < Z < 0.80
   10 IF(Z.GT.0.80) GOTO 20
      PROB2=(2.506628/Z)*EXP(-1.233701/ZSQ)
      GOTO 40
   20 IF(Z.GT.3.15) GOTO 30
C                 0.80 < Z < 3.15
      PROB2=1.-2.*(EXP(-2.*ZSQ)-EXP(-8.*ZSQ)+EXP(-18.*ZSQ))
      GOTO 40
C                 3.15 < Z
   30 PROB2=1.
   40 RETURN
      END
      SUBROUTINE NORM(DATA,ORIGIN,ISIZE,ANORM,IPOS)                  NORM    0
      DATNCR=DATA-ORIGIN                                             NORM    5
      PIONCR=DATNCR*ISIZE                                            NORM   10
      ANOW=PIONCR/ANORM                                              NORM   15
      IPOS=INT(ANOW+.555)+1                                          NORM   20
      RETURN                                                         NORM   25
      END                                                            NORM   30
      SUBROUTINE PRNCON(PARM,B,INP,ITYPE,TITLE)                      PRNC    0
      DIMENSION IPLOT(61,122),KYLAB(4),PARM(2,300),B(2),TITLE(4)     PRNC    5
      DATA KSTAR/'*'/,KGRID/'.'/,KTIC/'+'/,KBLK/' '/                 PRNC   10
      WRITE(6,10300)                                                 PRNC   15
      READ(6,11100) (KYLAB(I),I=1,4)                                 PRNC   20
   10 WRITE(6,11200)                                                 PRNC   25
      READ(6,11300) ICOL                                             PRNC   30
      IF (ICOL.NE.120.AND.ICOL.NE.60) GO TO 10                       PRNC   35
   20 WRITE(5,11400)                                                 PRNC   40
      READ(6,11500) IO                                               PRNC   45
      IF (IO.NE.'TTY') GO TO 30                                      PRNC   50
      IDEV=5                                                         PRNC   55
      GO TO 60                                                       PRNC   60
   30 IF (IO.NE.'DSK') GO TO 50                                      PRNC   65
   40 WRITE(6,11400)                                                 PRNC   70
      READ(6,11500)IDISK                                             PRNC   75
```

```
14080         IF (IDISK.LT.7.OR.IDISK.GT.63) GO TO 40                                PRNC  80
14090         OPEN(UNIT=IDISK,DEVICE='DSK',ACCESS='SEQOUT')                          PRNC  85
14100         IDEV=IDISK                                                             PRNC  90
14110         GO TO 60                                                               PRNC  95
14120      50 IF (IO.NE.'LPT') GO TO 20                                              PRNC 100
14130         IDEV=3                                                                 PRNC 105
14140         OPEN(UNIT=3,DEVICE='LPT',ACCESS='SEQOUT')                              PRNC 110
14150      60 DO 80 I=1,61                                                           PRNC 115
14160            IPLOT(I,1)=1                                                        PRNC 120
14170            DO 70 J=2,ICOL+2                                                    PRNC 125
14180               IPLOT(I,J)=KBLK                                                  PRNC 130
14190      70       CONTINUE                                                        PRNC 135
14200      80    CONTINUE                                                           PRNC 140
14210         DO 90 J=2,ICOL+2                                                      PRNC 145
14220            IPLOT(1,J)=KSTAR                                                   PRNC 150
14230            IPLOT(61,J)=KSTAR                                                  PRNC 155
14240      90    CONTINUE                                                          PRNC 160
14250         DO 110 I=2,60                                                         PRNC 165
14260            IPLOT(I,2)=KSTAR                                                   PRNC 170
14270            IPLOT(I,ICOL+2)=KSTAR                                              PRNC 175
14280            DO 100 J=7,ICOL-3,5                                                PRNC 180
14290               IPLOT(I,J)=KGRID                                                PRNC 185
14300     100       CONTINUE                                                        PRNC 190
14310     110    CONTINUE                                                           PRNC 195
14320         IF (ITYPE.NE.1) GO TO 140                                             PRNC 200
14330     120 WRITE(6,11600)                                                        PRNC 205
14340         WRITE(6,11600)                                                        PRNC 210
14350         READ(6,11700) YCUG                                                    PRNC 215
14360         IF (YCUG.LT.-1E8.OR.YCUG.GT.1E8) GO TO 120                            PRNC 220
14370         WRITE(6,11800)                                                        PRNC 225
14380         WRITE(6,11300)                                                        PRNC 230
14390         READ(6,11700) YSTEP                                                   PRNC 235
14400         DO 130 I=1,61,5                                                       PRNC 240
14410            IPLOT(I,2)=KTIC                                                    PRNC 245
14420     130    CONTINUE                                                          PRNC 250
14430         CALL LINPLT(IPLOT,YCUG,YSTEP,PARM,ICOL,INP,B)                         PRNC 255
14440         CALL PRNPLT(IPLOT,ITYPE,YCUG,ICYCLE,YSTEP,TITLE, KYLAB,IDEV,ICOL)     PRNC 260
14450         RETURN                                                                PRNC 265
```

```
14460      140 IF (IIYPE.NE.2) GO TO 150                                        PRNC 270
14470          WRITE(6,11200)                                                   PRNC 275
14480          RETURN                                                           PRNC 280
14490      150 WRITE(6,10600)                                                   PRNC 285
14500          WRITE(6,11700)                                                   PRNC 290
14510      160 WRITE(6,11900)                                                   PRNC 295
14520          READ(5,11750)YCHG                                                PRNC 300
14530          IF (YCHG.EQ.0.0) GO TO 160                                       PRNC 305
14540          IF (YCHG.LT.1E-15.OR.YCHG.GT.1E8) GO TO 150                      PRNC 310
14550      170 WRITE(6,11300)                                                   PRNC 315
14560          READ(5,11100)ICYCLE                                              PRNC 320
14570          IF (ICYCLE.LT.1.OR.ICYCLE.GT.5) GO TO 170                        PRNC 325
14580          IEXP=INT(ALOG10(YCHG))                                           PRNC 330
14590          CYCH=YCHG/10.**IEXP                                              PRNC 335
14600          CYCH=FLOAT(INT(CYCH))                                            PRNC 340
14610          ACYCH=CYCH*10.**IEXP                                             PRNC 345
14620      180 ITIER=60/ICYCLE                                                  PRNC 350
14630          DO 210 K=0,ICYCLE-1                                              PRNC 355
14640             ORG=CYCH                                                      PRNC 360
14650             KOUNT=0                                                       PRNC 365
14660             MAX=10.                                                       PRNC 370
14670             OFFAC=ALOG(CYCH)+ALOG(10.)                                    PRNC 375
14680      190    DO 200 AI=ORG,MAX                                             PRNC 380
14690             ALGTIC=ALOG(10.)+ALOG(AI)                                     PRNC 385
14700             CALL NORM(ALGTIC,OFFAC,ITIER,ALOG(10.),IROW)                  PRNC 390
14710             IROW=IROW+(ITIER*K)                                           PRNC 395
14720             IPLOT(IROW,2)=KTIC                                            PRNC 400
14730             IPLOT(IROW,1)=AI                                              PRNC 405
14740             KOUNT=KOUNT+1                                                 PRNC 410
14750      200    CONTINUE                                                      PRNC 415
14760             ORG=1                                                         PRNC 420
14770             MAX=CYCH                                                      PRNC 425
14780             OFFAC=ALOG(CYCH)                                              PRNC 430
14790             IF (KOUNT.LT.10) GO TO 190                                    PRNC 435
14800      210    CONTINUE                                                      PRNC 440
14810          CALL LOGPLT(IPLOT,ACYCH,ICYCLE,PAPM,INP,B,ICOL)                  PRNC 445
14820          CALL PENPLT(IPLOT,ITYPE,ACYCH,ICYCLE,YSTEP,TITLE, KYLAB,IDEV,ICOL)PRNC 450
```

```
14830          RETURN                                                          PRNC 455
14840    10110 FORMAT(/' ENTER Y-AXIS LABEL (24 CHARACTERS MAXIMUM)')          PRNC 460
14850    10100 FORMAT(4A5)                                                     PRNC 465
14860    10200 FORMAT(/' ENTER NUMBER OF COLUMNS FOR PLOT (60 OR 120)')        PRNC 470
14870    10310 FORMAT(I3)                                                      PRNC 475
14880    10400 FORMAT(/' ENTER OUTPUT DEVICE (TTY, DSK, OR LPT)')              PRNC 480
14890    10510 FORMAT(A3)                                                      PRNC 485
14900    10600 FORMAT(/' ENTER Y-AXIS ORIGIN')                                 PRNC 490
14910    10700 FORMAT(G)                                                       PRNC 495
14920    10800 FORMAT(/' ENTER Y-AXIS STEP SIZE (UNITS/5 LINES OF PLOT,')      PRNC 500
14930    10900 FORMAT(' (CANNOT BE ZERO)')                                     PRNC 505
14940    11000 FORMAT(/' ENTER NUMBER OF CYCLES FOR LOG PLOT (INTEGER,1-5)')   PRNC 510
14950    11100 FORMAT(I1)                                                      PRNC 515
14960    11200 FORMAT(/' UNRECOGNIZED PLOT TYPE ; ERROR IN MAIN')              PRNC 520
14970    11300 FORMAT(' 60 LINES OF PLOT)')                                    PRNC 525
14980    11400 FORMAT(/' ENTER DISK FILE NUMBER (7-63)')                       PRNC 530
14990    11500 FORMAT(I2)                                                      PRNC 535
15000    11600 FORMAT(/' BETWEEN -1E8 AND 1E8')                                PRNC 540
15010    11700 FORMAT(/' BETWEEN 1E-15 AND 1E8')                               PRNC 545
15020          END                                                             PRNC 550
15030          SUBROUTINE PRNPLT(IPLOT,ITYPE,YORG,ICYCLE,YSTEP,TITLE, KYLAB,IDEV,PRNP   0
15040         1 ICOL)                                                          PRNP   5
15050          DIMENSION IPLOT(61,122),TITLE(5),KYLAB(4)                       PRNP  10
15060          IF (ICOL.EQ.120) GO TO 10                                       PRNP  15
15070          WRITE(IDEV,10300) (TITLE(I),I=1,4)                              PRNP  20
15080          WRITE(IDEV,10100) (KYLAB(I),I=1,4)                              PRNP  25
15090          GO TO 20                                                        PRNP  30
15100       10 WRITE(IDEV,10200) (TITLE(I),I=1,4)                              PRNP  35
15110          WRITE(IDEV,10300) (KYLAB(I),I=1,4)                              PRNP  40
15120       20 CONTINUE                                                        PRNP  45
15130          IF (ITYPE.NE.1) GO TO 60                                        PRNP  50
15140          IWLAB=5                                                         PRNP  55
15150          YLAB=YORG+(12*YSTEP)                                            PRNP  60
15160          DO 50 I=61,1,-1                                                 PRNP  65
15170             IF (IWLAB.NE.5) GO TO 30                                     PRNP  70
15180             IWLAB=0                                                      PRNP  75
15190             WRITE(IDEV,10400)YLAB,(IPLOT(I,J),J=2,ICOL+2)                PRNP  80
```

```
                YLAB=YLAB-YCYCLE                                        PBNE  85
                GO TO 40                                                PBNE  90
   30           WRITE(IDEV,10500) (IPLOT(I,J),J=2,ICOL+2)               PBNE  95
   40           CONTINUE                                                PBNE 100
                IWLAB=IWLAB+1                                           PBNE 105
   50           CONTINUE                                                PBNE 110
            GO TO 110                                                   PBNE 115
   60       CONTINUE                                                    PBNE 120
            ICYCLE=6 /ICYCLE                                            PBNE 125
            IWLAB=ICYCLE                                                PBNE 130
            YLAB=YORG*10.**ICYCLE                                       PBNE 135
            DO 100 I=61,1,-1                                            PBNE 140
                IF (IWLAB.NE.ICYCLE) GO TO 70                           PBNE 145
                WRITE(IDEV,10400) YLAB,(IPLOT(I,J),J=2,ICOL+2)          PBNE 150
                IWLAB=0                                                 PBNE 155
                YLAB=YLAB/10.                                           PBNE 160
                GO TO 90                                                PBNE 165
   70           IF (IPLOT(I,1).EQ.0) GO TO 80                           PBNE 170
                WRITE(IDEV,10600) (IPLOT(I,J),J=1,ICOL+2)               PBNE 175
                GO TO 90                                                PBNE 180
   80           WRITE(IDEV,10500) (IPLOT(I,J),J=2,ICOL+2)               PBNE 185
   90           CONTINUE                                                PBNE 190
                IWLAB=IWLAB+1                                           PBNE 195
  100       CONTINUE                                                    PBNE 200
  110       IF (ICOL.EQ.120) GO TO 120                                  PBNE 205
            WRITE(IDEV,10700)                                           PBNE 210
            WRITE(IDEV,10800)                                           PBNE 215
            GO TO 130                                                   PBNE 220
  120       WRITE(IDEV,10900)                                           PBNE 225
            WRITE(IDEV,11000)                                           PBNE 230
  130       CLOSE(UNIT=IDEV)                                            PBNE 235
            RETURN                                                      PBNE 240
10000 FORMAT(20X,A5)                                                    PBNE 245
10100 FORMAT(5X,4A5)                                                    PBNE 250
10200 FORMAT(50X,A5)                                                    PBNE 255
10300 FORMAT(5X,4A5)                                                    PBNE 260
10400 FORMAT(1X,1PE10.3,121A1)                                          PBNE 265
```

```
15570   10590 FORMAT(11X,121A1)                                                 PRNP 270
15580   10600 FORMAT(9X,I2,121A1)                                               PRNP 275
15590   10700 FORMAT(10X,'.14',2X,'.62',2X,'2.3',2X,'6.7',3X,'16', 3X,'31',3X,  PRNP 280
15600        1 '50',3X,'69',3X,'84',3X,'93',2X, '97.7',1X,'99.4',1X,'99.9')     PRNP 285
15610   10800 FORMAT(25X,'CUMULATIVE PROBABILITY (%)')                          PRNP 290
15620   10900 FORMAT(10X,'.14',7X,'.62',7X,'2.3',7X,'6.7',7X,'16', 8X,'31',8X,  PRNP 295
15630        1 '50',8X,'69',8X,'84',8X,'93',7X, '97.7',6X,'99.4',5X,'99.9')     PRNP 300
15640   11000 FORMAT(57X,'CUMULATIVE PROBABILITY (%)')                          PRNP 305
15650         END                                                               PRNP 310
15660         SUBROUTINE UERTST (IER,NAME)                                      UERT   0
15670   C                                                                       UERT   5
15680   C-UERTST----------------LIBRARY 1-----------------------------------UERT  10
15690   C                                                                       UERT  15
15700   C   FUNCTION              - ERROR MESSAGE GENERATION                    UERT  20
15710   C   USAGE                 - CALL UERTST(IER,NAME)                       UERT  25
15720   C   PARAMETERS   IER      - ERROR PARAMETER. TYPE + N  WHERE            UERT  30
15730   C                             TYPE= 128 IMPLIES TERMINAL ERROR          UERT  35
15740   C                                   64 IMPLIES WARNING WITH FIX         UERT  40
15750   C                                   32 IMPLIES WARNING                  UERT  45
15760   C                             N   = ERROR CODE RELEVANT TO CALLING ROUTINEUERT  50
15770   C                 NAME    - INPUT VECTOR CONTAINING THE NAME OF THE     UERT  55
15780   C                             CALLING ROUTINE AS A SIX CHARACTER LITERAL UERT  60
15790   C                             STRING.                                   UERT  65
15800   C   LANGUAGE              - FORTRAN                                     UERT  70
15810   C-----------------------------------------------------------------------UERT  75
15820   C   LATEST REVISION       - JANUARY 1, 1974                             UERT  80
15830   C                                                                       UERT  85
15840   C                                                                       UERT  90
15850         DIMENSION ITYP(5,4),IBIT(4)                                       UERT  95
15860         INTEGER NAME(3)                                                   UERT 100
15870         INTEGER WARN,JARN,TERM,PRINTS                                     UERT 105
15880         EQUIVALENCE (IBIT(1),WARN),(IBIT(2),WARN),(IBIT(3),TERM)          UERT 110
15890         DATA ITYP /'WARN','ING ','    ','    ','    ', 'WARN','ING(',   UERT 115
15900        1 'WITH',' FIX',')   ', 'TERM','INAL','    ','    ','    ', 'NON-',UERT 120
15910        1  'DEFI','NED ','    ','    '/                                   UERT 125
15920         IBIT(1)=32                                                        UERT 130
15930         IBIT(2)=64                                                        UERT 135
15940         IBIT(3)=128                                                       UERT 140
```

```
      IBIT(4)=8                                                          UERT 145
      DATA PRINT / 6/                                                    UERT 150
      IER2=IER                                                           UERT 155
      IF (IER2.GT.WARN) GO TO 10                                         UERT 160
C                                          NON-DEFINED                   UERT 165
      IER1=4                                                             UERT 170
      GO TO 40                                                           UERT 175
   10 IF (IER2.LT.TERM) GO TO 20                                         UERT 180
C                                          TERMINAL                      UERT 185
      IER1=3                                                             UERT 190
      GO TO 40                                                           UERT 195
   20 IF (IER2.LT.WARN) GO TO 30                                         UERT 200
C                                          WARNING(WITH FIX)             UERT 205
      IER1=2                                                             UERT 210
      GO TO 40                                                           UERT 215
C                                          WARNING                       UERT 220
   30 IER1=1                                                             UERT 225
C                                          EXTRACT 'N'                   UERT 230
   40 IER2=IER2-IBIT(IER1)                                               UERT 235
C                                          PRINT ERROR MESSAGE           UERT 240
      WRITE(PRINT,10000) (ITYP(I,IER1),I=1,5),NAME,IER2,IER              UERT 245
      RETURN                                                             UERT 250
10000 FORMAT(' *** I M S L(UTTEST) ***  ',5A4,4X,3A2,4X,I2, ' (IER = ',  UERT 255
     1 I3,')')                                                           UERT 260
      END                                                                UERT 265
```

65/16

ORNL-5967
Dist. Category UC-11, UC-41

INTERNAL DISTRIBUTION

1. C. F. Baes III
2. L. W. Barnthouse
3. J. J. Beauchamp
4. M. A. Bell
5. R. O. Chester
6. W. G. Cottrell
7. K. E. Cowser
8. W. G. Craddick
9. W. Davis, Jr.
10. K. F. Eckerman
11-20. D. E. Fields
21. G. Holton
22. S. V. Kaye
23. G. D. Kerr
24. C. W. Miller
25. B. D. Murphy
26. T. W. Oakesy
27 S. Ohr
28. M. R. Patterson
29. D. C. Parzyck
30. M. R. Patterson
31 P. S. Rohwer
32. G. Suter
33. M. Yildiran
34 Central Research Library
35. ORNL Y-12 Technical Library
36. Laboratory Records
37. Laboratory Records - RC
38. ORNL Patent Office

EXTERNAL DISTRIBUTION

39. Larry D. Carpenter, Route 2, Box 45-A, Watertown, TN 37184
40. P. R. Jaffe, Department of Civil Engineering, Princeton University, Princeton, NJ 08544
41-50. S. E. Kurtz, 2821 McCain Rd., Jackson, MI 49203
51. Mei-Kao Liu, Systems Applications, Inc., 950 Northgate Drive, San Rafael, CA 94903
52. D. Lynn Shaeffer, The BDM Corporation, 1801 Randolph Road, SE, Albuquerque, NM 87106
53. Office of Assistant Manager for Energy Research and Development, Department of Energy, Oak Ridge Operations Office, Oak Ridge, TN 37830
54-363. Given distribution as shown in TID-4500 under UC-11 and UC-41